
GAZİ ÜNİVERSİTESİ
Gazi
1926

EMOTION RECOGNITION FROM FACIAL EXPRESSIONS USING DEEP LEARNING APPROACHES IN INFORMATION TECHNOLOGIES

Ahmed Adnan Hameed QUTUB

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE
IN COMPUTER ENGINEERING DEPARTMENT

GAZİ UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

JANUARY 2024

ETHICAL STATEMENT

I at this moment declare that in this thesis study, I prepared the thesis writing rules of Gazi University Graduate School of Natural and Applied Sciences;

- All data, information, and documents presented in this thesis have been obtained within the scope of academic rules and ethical conduct,
- All information, documents, assessments, and results have been presented by scientific ethical conduct and moral rules,
- All material used in this thesis that is not original to this work has been exhaustively cited and referenced,
- No change has been made in the data used,
- The work presented in this thesis is original,

or else, I admit all loss of rights to be incurred against me.

Ahmed Adnan Hameed QUTUB

02/01/2024

EMOTION RECOGNITION FROM FACIAL EXPRESSIONS USING DEEP LEARNING APPROACHES IN INFORMATION TECHNOLOGIES

(Master Thesis)

Ahmed Adnan Hameed QUTUB

GAZİ UNİVERSİTY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

January 2024

ABSTRACT

Automated emotion recognition plays a pivotal role in various fields reliant on emotional responses, such as advertising, technology, and human-robot interaction, particularly within the information technology (IT) sector. For machines to be seamlessly integrated into our daily lives, they must be able to understand another person's emotional state from their point of view. Previous studies on computer vision for emotion recognition mainly focus on evaluating facial expressions and classifying them into six basic emotions such as fear, anger, sadness, surprise, happiness, and natural (normal). This study introduces multiple deep Convolutional Neural Network (CNN) models trained on distinct datasets, shedding light on the limitations of these datasets in capturing the depth of human emotions. The CNN model specifically proposed for emotion recognition tasks was designed with reference to the FER2013 and RAF-DB datasets. The trial-and-error approach determines the optimal learning rate and other data augmentation parameter values for all applied CNN models. A genetic algorithm was applied together with CNN for hyperparameter optimization. The genetic algorithm has been employed to enhance the performance, given the multitude of activation functions, layer types, and hyperparameters involved in designing a convolutional neural network. After that more complex models, including ResNet18, VGGNet16, VGGNet19, and EfficientNet-B0, were adapted to the problem, and the performance of deep learning approaches were increased. The performances of all models were tested on FER2013 and RAF-DB datasets. The VGGNet model is too complex and large compared to other models due to its original 138M parameters. However, with the customization approach used, the number of parameters was reduced to 45.2M. When the experimental results were examined, it was observed that among all models, the VGGNet19 model achieved the most appropriate results with a test accuracy of 71.02% on FER2013. In tests on the RAF-DB dataset, the VGGNet19 model reached 85.87% test accuracy; the ResNet18 model outperformed all other models with a test performance of 86.02%. These findings showed that automatic emotion recognition systems can be effectively developed with the proposed customization approach. It has been emphasized that it is necessary to recommend advanced systems for applications in information technologies in future studies.

Science Code	:	92432
Key Words:	:	Information technologies, emotion recognition, deep learning, genetic algorithm, VGGNet, Resnet, EfficientNet.
Page Number	:	85
Supervisor	:	Asst. Prof. Dr. Yılmaz ATAY

BİLİŞİM TEKNOLOJİLERİNDE DERİN ÖĞRENME YAKLAŞIMLARI KULLANILARAK YÜZ İFADELERİNDEN DUYGU TANIMA

(Yüksek Lisans Tezi)

Ahmed Adnan Hameed QUTUB

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Ocak 2024

ÖZET

Otomatik duygu tanıma; reklamcılık, teknoloji ve insan-robot etkileşimi gibi duygusal tepkilere dayalı çeşitli alanlarda, özellikle de bilgi teknolojisi (BT) sektöründe çok önemli rol oynar. Makinelerin günlük hayatımıza sorunsuz bir şekilde entegre olabilmesi için başka bir kişinin duygusal durumunu kendi bakış açısından anlayabilmeleri gerekir. Duygu tanıma için bilgisayarlı görme üzerine yapılan önceki çalışmalar ağırlıklı olarak yüz ifadelerini değerlendirmeye ve bunları korku, öfke, üzüntü, sürpriz, mutluluk ve normal türlerinde altı temel duyguya ayırmaya odaklanır. Bu çalışma, farklı veri kümeleri üzerinde eğitilmiş çok sayıda derin Evrişimli Sinir Ağı (CNN) modelini tanıtmakta ve bu veri kümelerinin insan duygularının derinliğini yakalamadaki sınırlamalarına ışık tutmaktadır. Duygu tanıma görevleri için özel olarak önerilen CNN modeli, FER2013 ve RAF-DB veri kümeleri referans alınarak tasarlanmıştır. Deneme yanılma yaklaşımı, uygulanan tüm CNN modelleri için en uygun öğrenme oranını ve diğer veri artırma parametre değerlerini belirler. Hiperparametre optimizasyonu için CNN ile birlikte genetik algoritma uygulanmıştır. Evrişimli bir sinir ağının tasarımında yer alan çok sayıda aktivasyon fonksiyonu, katman türü ve hiperparametre göz önüne alındığında, performansı artırmak için genetik algoritma kullanılmıştır. Ardından ResNet18, VGGNet16, VGGNet19 ve EfficientNet-B0 gibi daha karmaşık modeller soruna uyarlanarak derin öğrenme yaklaşımlarının performansı artırıldı. VGGNet modeli, orijinal 138M parametreleri nedeniyle diğer modellere göre daha karmaşık ve büyüktür. Ancak kullanılan özelleştirme yaklaşımıyla parametre sayısı 45,2 milyona düşürülmüştür. Deneysel sonuçlar incelendiğinde, tüm modeller arasında VGGNet19 modelinin FER2013 üzerinde %71,02 test doğruluğu ile en uygun sonuçları elde ettiği gözlemlenmiştir. RAF-DB veri seti üzerindeki testlerde ise VGGNet19 modeli %85,87 test doğruluğuna ulaşırken; ResNet18 modeli %86,02 test performansıyla diğer tüm modelleri geride bırakmıştır. Bu bulgular, önerilen özelletirme yaklaşımı ile otomatik duygu tanıma sistemlerinin etkin bir şekilde geliştirilebileceğini göstermiştir. Gelecek çalışmalarda bilgi teknolojileri alanındaki uygulamalar için ileri düzey sistemlerin önerilmesinin gerekliliği vurgulanmıştır.

1. Giriş

Derin öğrenme, son yıllarda önemli ilerlemeler kaydetmiş ve sahne tanımlamadan nesne tespitine kadar çeşitli bilgisayar görüşü alanlarında çığır açan başarılar göstermiştir [1-3]. Derin sinir ağı tabanlı tekniklerin kullanımı, duygu tanıma verilerini yorumlama ve anlama yeteneklerimizi önemli ölçüde artırmıştır. Bu tür çabalar sayesinde duygu tanımada ortaya çıkan zorlukları ele alma konusunda uygun bir strateji oluşturmak için çeşitli derin öğrenme

mimarilerinden yararlanılmaktadır. İletişimin temel bir unsuru olan jestler, bilgisayarlı görü ve duygu tanıma gibi alanlarda önemli rol oynar [4]. Bu çalışma, basit verilere uygulanan temel stratejilerden daha karmaşık veri kümeleriyle başa çıkmak için sofistike bir yaklaşım geliştirmeyi hedeflemektedir. Ayrıca duyguların karmaşık yapısına daha derinlemesine bakabilmeyi ve bilgisayarlı görünün karmaşık duygusal ifadeleri ele almadaki dikkate değer uygulamalarını keşfetmeyi amaçlamaktadır. Duygu tanımanın karmaşıklığı, duyguların genellikle hassas olmasından ve yorumlanmasının zorluğundan kaynaklanır. Örneğin, bir gülümseme mutluluğa işaret edebilir ancak aynı zamanda alaycılık veya küçümsemeyi de ifade edebilir. Benzer şekilde, bir kaş çatma üzüntüyü gösterebilir ancak bununla birlikte öfke veya doyumsal yoksunluk anlamına da gelebilir. Bu karmaşıklıklarla başa çıkabilmek amacıyla bilgisayarlı görü araştırmacıları, yüz ifadelerini ve diğer sözsüz ipuçlarını doğru bir şekilde tanımlayabilmek için çeşitli teknikler geliştirmişlerdir. Böylece duygu tanıma sistemlerinin doğruluğunu ve güvenilirliğini artırmak için yeni teknikler ve algoritmalar geliştirmek mümkündür. Bu, sağlık, eğitim ve eğlence gibi geniş yelpazeye sahip alanlar için önemli sonuçlar doğurmaktadır.

Problem tanımı

Tez çalışmasındaki problemi ele alırken temel amaç, basit bir yaklaşımdan karmaşık bir yaklaşıma doğru sistemli bir model geliştirmektir. Bu süreçte, basit verilerden daha zorlu veri kümelerine doğru aşamalı bir öğrenme yaklaşımı kullanılarak geçiş yapılması hedeflenmektedir. Problemi daha derinlemesine inceleyebilmek için artımlı bir öğrenme yaklaşımı benimsenmektedir. Ayrıca bilgisayarlı görü alanındaki yöntemlerin karmaşık görevlere etkili bir şekilde nasıl uygulanabileceği konusunda araştırmaların yapılması amaçlanmıştır.

Katkı

Araştırma kapsamında etkili bir yaklaşım oluşturabilmek için FER2013 [5,6] ve RAF-DB [7] adlı iki açık kaynak veri kümesi kullanılmaktadır. Bu veri kümeleri, jest tanıma problemine genel bir bakış açısı kazandırmaktadır. Bu konudaki temel katkılarımız aşağıda özetlenmiştir:

• Başlangıç modelinin oluşturulması: Tez çalışmasındaki ilk model, sonraki gelişmelere bir referans olarak hizmet eder. Transfer öğreniminin ve yaygın derin öğrenme mimarilerinin etkisini anlamamıza yardımcı olur.

•Çeşitli derin öğrenme modellerinin kullanılması: ResNet18 [8], VGGNet16[9], VGGNet19 [9] ve EfficientNet-B0 [10] gibi farklı derin öğrenme modelleri, her iki veri kümesinde makul doğrulukla sağlam bir model oluşturmamıza katkı sağlar.

• Genetik algoritmanın uygulanması: Bu algoritma, yeni ve etkili bir CNN model mimarisi tasarlamamıza yardımcı olur. Popülasyon modelleri için uygunluk fonksiyonu olarak tasarlanan modellerin Çapraz Entropi kaybı seçilir.

•Kullanılan modellerin test setinde değerlendirilmesi: FER2013 veri kümesinde, VGGNet19 modeli %71,02 test doğruluğu ile en iyi sonuçları elde ederken; 138M parametreli orijinal VGGNet modeli aşırı karmaşık olmasına rağmen bu özelleştirme ile parametre sayısı 45,2M'ye düşürülerek daha adil bir karşılaştırma yapılır. RAF-DB veri kümesinde ise VGGNet19 %85,85 test doğruluğuna ulaşırken; ResNet18 diğer tüm modelleri %86,02 test

doğruluğu ile geride bırakır. Bu katkılar, bilgisayarlı görü alanında karmaşık görevler için otomatik jest tanıma sistemlerinin geliştirilmesinde çalışmanın önemini vurgular.

Sınırlama

Otomatik yüz ifadesi tanıma sistemlerinde, yüz ifadesi tanıma alanı doğal dil işleme (NLP) fikriyle birleştirilerek daha boyutlu hale getirilebilir. Gelecekteki kapsam uygulandığında e-sağlık sisteminde ve sağlık hizmetlerinin sunumunda daha etkili rol oynayabilir. İnsan dilini yüz ifadeleriyle birleştirerek; bir insanın duygusunu tahmin etme, gerçekten büyük bir gelişme sunabilecektir. Bu durum, sisteme hareketleri daha doğru tahmin etme konusunda faydalı bilgiler sağlayabilecektir. Önerilen yaklaşım, sadece yüz ifadesine dayanmakta ve bu çalışmanın temel sınırlamasını ifade etmektedir. Problem, yalnızca yüz ifadelerini sunan FER2013 ve RAF-DB veri setlerinde ele alınmaktadır. İnsan yüz ifadeleri hakkında bilgi toplayabilmek için CNN mimarileri kullanılmaktadır.

2. Yöntem & önerilen yaklaşımlar

Önerilen yaklaşımın çerçevesi, bu bölümde kısaca açıklanmaktadır. Çalışmada iki açık veri seti kullanılmaktadır. Bunlar FER2013 [5 , 6] ve RAF-DB [7] veri setleridir. Bunlarla ilgili aynı iş süreçleri her bir veri kümesine bağımsız olarak uygulanır. Başlangıçta, veri kümeleri yüklenir. Doğrulama ve eğitim kümelerine bölünür. Her veri kümesi için önceden tanımlanmış bir test kümesi bulunur. Eğitim sırasında gerektiğinde veri kümesini işleyebilen ve örnekleri veri kümeleri sağlam bir veri yükleyici kullanılır. Aşırı öğrenme problemini azaltmak ve eğitim veri kümelerinin farklı özelliklerini keşfedebilmek için çeşitli veri artırma teknikleri bu süreçte kullanılır. Bu bölüm, uygulanan veri artırma iş sürecini ve kullanılan her hiper parametreyi detaylandırmaktadır. Makul doğrulukla sağlam bir model oluşturabilmek için çeşitli derin öğrenme modelleri ele alınmıştır. Bunlar ResNet18, VGGNet16, VGGNet19 ve EfficientNet-B0 modelleridir. Ayrıca başlangıç modeli, sonraki gelişmeler için bir referans sağlar. Bu, transfer öğreniminin ve yaygın derin öğrenme mimarilerinin etkilerini anlamamıza olanak tanır. Daha derinlemesine keşfedebilmek için genetik algoritma, bir CNN model mimarisi tasarlamak için kullanılır. Popülasyon modellerinin uygunluk fonksiyonu, tasarlanan modellerin Çapraz Entropi kaybı olarak seçilir. Ardından, modeller eğitilir ve test kümesinde değerlendirilir. Bu modeller ve yaklaşımlar arasında test kümesinde elde edilen doğruluk temel alınarak; kısa bir karşılaştırma sunulur. Her model için hassasiyet, kesinlik ve F1 skoru gibi ek metrikler hesaplanır. Ayrıca önerilen modellerde bulunan hataların türlerini daha iyi anlamak için her eğitilmiş modelin karışıklık/hata matrisi hesaplanır. Bu bölüm, her bir modeli ayrı ayrı daha ayrıntılı olarak ele alır ve modellerin mimarisini açıklar. Bu metriklere dayalı olarak ilgili modelleri karşılaştırabilmek için kullanılan her değerlendirme metriği detaylandırılır ve önerilen modellerin bilinmeyen veriler üzerindeki performansı incelenir.

Evrişimli sinir ağı ve derin öğrenme yaklaşımları

Evrişimli Sinir Ağı (CNN) [11] görüntü tanıma, sınıflandırma, eşleştirme ve benzer görevlerde yaygın olarak kullanılan güçlü bir derin öğrenme algoritmasıdır. Geleneksel modellerin aksine CNN, minimal ön işleme gerektirir ve manuel özelliğe ihtiyaç duymadan otomatik olarak öğrenir. Bu yaklaşım, küçük parçalara bölünen görüntüleri işleyerek karmaşık desenleri algılama yeteneğini artırır. CNN'nin temel bileşenleri arasında giriş katmanı, çıkış katmanı ve evrişimli katmanlar, havuzlama katmanları (maksimum ve

ortalama), tam bağlantılı katmanlar (FC) ve normalleştirilmiş katmanlar bulunur. CNN, girdi görüntülerinden özellikleri çıkarmak için ağırlık dizileri olan filtreleri (çekirdekleri) kullanır. Her katmanda farklı aktivasyon fonksiyonları uygulanır ve bu da doğrusallığı dışlayarak daha karmaşık verilere uyum sağlar. CNN işlemi sırasında yükseklik ve genişlik azalırken kanal sayısı artar. Sonuç olarak sütun matrisi, çıktıyı tahmin etmek için kullanılır.

Evrişimli Sinir Ağları, çeşitli görsel problemlerde uygun çözümler üreterek etkinliğini kanıtlamıştır. Her evrişim katmanı, girdi kanalları içinde mekansal bağlantı desenlerini ifade eden bir dizi filtre kullanır. CNN'ler, evrişimli katmanları (non-linear activation functions) ve (down-sampling) operatörlerini sırasıyla birleştirerek hiyerarşik resim temsilleri oluşturabilir ve hem mekansal hem de kanal bazlı bilgiyi yakalayabilir. Daha sağlam temsilleri vurgulayan, böylece performansı artıran özgül görevler için temel görüntü özelliklerini vurgulayan ve merkezi bir tema olan bilgisayarlı görü araştırmalarında önemli rol oynamaktadır. Son dönemdeki gelişmeler, özellikler arasındaki mekansal ilişkileri yakalayarak temsilleri geliştirmek için öğrenme mekanizmalarını CNN'lere entegre etmeyi önermektedir. Literatürde popülerleştirilen dikkate değer teknikler, ağ modüllerine çok ölçekli işlemleri dahil etmeyi içerir ve performansı artırır.

Genetik algoritma

Genetik algoritma (GA); mutasyon, çaprazlama ve seçim gibi biyolojiden esinlenen kavramları kullanarak optimal çözümler üretmek için uyarlanabilir sezgisel bir arama yöntemidir. GA'nın çözüm arama süreci; mimari varyasyonları olan CNN'leri temsil eden bireylerin rastgele başlatılmasıyla başlar [12]. Her üyenin uygunluk durumu, belirli görüntü sınıflandırma görevlerindeki performanslara bağlı olarak deterministik bir uygunluk fonksiyonu tarafından ölçülerek değerlendirilir. Değerlendirmenin ardından seçim prosedürü, en yüksek uygunluğa sahip bireyleri belirler. Çaprazlama ve mutasyon operatörleri daha sonrasında seçilen ebeveynlerden yeni bireyler (yavrular) üretir. Uygunluk fonksiyonu, seçim sonrası yavruları değerlendirir ve orijinal popülasyonun en iyi üyeleri ile onların soyu, sabit bir boyutta yeni bir popülasyon oluşturur. GA, belirlenmiş bir nesil sayısı boyunca devam eder. Sonlandırma kriteri olarak maksimum nesil sayısı belirlenir (örneğin, 20 nesil). Kullanılan operatörler, GA'nın yakınsama yeteneği üzerinde büyük bir etkiye sahiptir (yani seçim, çaprazlama ve mutasyon). Operatörler, genetik çeşitliliği korumak (mutasyon operatörü), eski çözümleri birleştirerek yeni çözümler üretmek (çaprazlama operatörü) ve çözümler arasında seçim yapmak için kullanılır. Aşağıda her operatör için kısa açıklamalar sunulmuştur.

- Seçim: Bu operatör, her nesilde yeni yavrular üretecek popülasyon üyelerini belirler. Farklı yaklaşımlar aracılığıyla farklı çözümler tercih eden çeşitli stratejiler bulunmaktadır.
- Çaprazlama: Seçilen iki bireyin parçalarını birleştirerek yeni bir birey oluşturan çaprazlama operatörünün, seçilen ebeveynler uygunsa daha iyi bir birey üretme olasılığı daha yüksektir. Tek nokta çaprazlama gibi farklı stratejiler, ebeveyn çözümünün parçalarını birleştirmeyi içerir.
- Mutasyon: GA'nın yerel minimuma yakınsamasını önler ve bireyler arasında genetik çeşitliliği teşvik eder. Bireyi önemli ölçüde değiştirebilir. Böylece bireyler arasındaki yakınlığı bozar ve erken yakınsamayı engeller. Basit bir permütasyon (örneğin, CNN'lerde iki katmanın yer değiştirmesi) gibi çeşitli mutasyon teknikleri uygulanabilir.

VGGNet16

Bu model, 1000 sınıfa bölünen ve 14 milyondan fazla görüntü içeren ImageNet veri kümesinde %92.7'lik oranla en iyi beş test doğruluk başarısına ulaşmıştır [9]. VGG16, ILSVRC-2014'e sunulan ve iyi bilinen modellerden biri oldu. AlexNet'in daha büyük boyutlu çekirdek filtrelerini (sırasıyla ilk ve ikinci evrişim katmanlarında 11 ve 5) birkaç 3x3 boyutlu çekirdek filtresiyle başarıyla değiştirilerek bu sonuca ulaşıldı. 224x224 boyutunda tanımlanmış bir RGB görüntüsü, Cov1 katmanı için girdi olarak kullanılır. Görüntü, 3x3 boyutlu bir alanda sol/sağ, yukarı/aşağı ve merkez kavramlarını yakalayan bir dizi evrişim (conv.) filtresi ile işlenir. Girdi kanallarının doğrusal dönüşümü olarak hizmet eden 1x1 boyutlu evrişim filtreleri, bir konfigürasyonda da kullanılır. Evrişim katmanının girişine uygulanan mekansal dolgu, evrişimden sonra mekansal çözünürlüğün korunmasını sağlamak üzere tasarlanmıştır. Yani 3x3 boyutlu conv. katmanları için dolgu 1 pikseldir ve evrişim adımı 1 piksel olarak sabittir. Beş üst katman, maksimum havuzlama katmanlarıdır ve 2x2 piksel çerçevesinde iki adımlı mekansal havuzlama yapar. Farklı tasarımlarda değişen derinliklere evrişim katmanlarının ardından üç Tam Bağlantılı (FC) katman bulunur: Bunlardan ilk ikisi sırasıyla 4096 kanala sahiptir. Üçüncüsü ise 1000 yollu ILSVRC sınıflandırması yapar ve 1000 nörona sahiptir. Son katman, softmax katmanıdır. Her ağda, tam bağlantılı katmanların konfigürasyonları aynıdır. Gizli katmanların tümünde aktivasyon fonksiyonu olarak Doğrusal Düzeltme (ReLU) kullanılır. Ayrıca ILSVRC veri kümesinde performansı artırmadan bellek kullanımı ile hesaplama süresini artıran ve Local Response Normalization (LRN) içermeyen tek ağın dikkate alınması gerektiğini belirtmek gerekir.

VGGNet19

Bu modeli görüntü sınıflandırma için kullanılan evrişimli sinir ağı olan VGGNet'in bir mimarisidir. Toplamda 19 katmandan oluşur [9]. Bunların 16'sı evrişim katmanı, 3'ü tam bağlantılı katmandır. Evrişim katmanları iki veya üçlü gruplar halinde düzenlenmiştir ve aralarında maksimum havuzlama katmanları bulunmaktadır. VGGNet19, karmaşık özellikleri görüntülerde yakalayabilen çok derin bir mimariye sahiptir. Model, ağdaki her katmanın giriş ve çıkış boyutlarını, her evrişim katmanında kullanılan filtre sayısını ve çekirdek boyutlarını vurgular. Genel olarak VGGNet19, çeşitli bilgisayarlı görü süreçlerinde yaygın olarak kullanılan güçlü bir CNN mimarisi olarak bilinir ve birçok uygulamada öne çıkar.

ResNet18

ResNet18 [8], derin ağları daha verimli bir şekilde eğitebilmek için uygun bir yaklaşım sunar. Geleneksel derin ağlar, derin modelleri iyi ve verimli bir şekilde eğitme konusunda zorluklarla karşılaşır. ResNet'in getirdiği yenilik ise bir ağdaki katmanların nasıl öğrendiğini tekrar düşünerek artık fonksiyonlarına odaklanmaktadır. Daha derin ağlar, öğrenme açısından faydalı olsa da eğitimi zorlaşır ve hızlı bir düşüşün ardından doğruluk doygunluğu yaşar. Residual learning, bu zorluğu aşarak daha derin ağların eğitimini mümkün kılar ve doğruluk düşüşü problemini önler. Basit ağlarda istenen eşleme hemen birkaç katmanın bir araya getirilmesiyle öğrenilirken residual ağlardaki katmanlar artık eşlemeyi öğrenmek üzere bir araya getirilir.

ResNet-18'in orijinal mimarisi, toplam 18 katmandan oluşur. Bu katmanlar arasında sınıflandırma için 17 evrişim katmanı, 1 tam bağlantılı katman ve 1 (softmax katmanı) bulunur. Tasarım, çıktı özellik haritasının evrişim katmanlarının 3x3 filtreleriyle aynı

boyutta olduğunda, katmanların aynı sayıda filtreye sahip olmasını sağlar. Ancak çıktı özellik haritası yarıya indirilirse katmanlardaki filtre sayısı ikiye katlanır. İki adımlı evrişim katmanları veriyi küçültmek için kullanılır. Son iki katman ise ortalama havuzlama, tam bağlantılı bir katman ve bir softmax katmanından oluşur. Ağın her seviyesinin arasına kısa yol bağlantıları eklenir. İki tür bağlantı kullanılır. İlk tür, katmanın giriş ve çıkış boyutları aynı olduğunda kimlik eşlemesi yapar ve kesikli çizgilerle temsil edilir. İkinci tür, boyut genişlemesi için bağlantılar kullanır ve noktalı çizgilerle gösterilir. Ayrıca ikinci tür, daha büyük boyutlar için sıfır dolgulu kimlik eşlemesini korur ve bir adımlık kaydırma kullanır.

EfficientNet

Bu model [10], veri bilimi alanında daha az parametre ile daha yüksek sınıflandırma doğruluğu elde etmek amacıyla genişlik ve derinlik arasındaki ilişkiyi inceler ve pratik bir yöntem geliştirir. EfficientNetB0 ile EfficientNetB7 olarak adlandırılan yedi model, ImageNet veri kümesi ile değerlendirildiğinde parametre sayısı ve Top-1 doğruluk açısından üstün bir performans sergilemektedir. EfficientNet'in ölçeklenebilirliği, basit ancak son derece etkili bir bileşik katsayı kullanılarak gerçekleştirilir. Geleneksel yöntemlerin aksine genişlik, derinlik ve çözünürlük gibi ağ boyutlarını bağımsız bir şekilde ölçeklendirmek yerine, EfficientNet her boyutu belirli bir dizi ölçekleme katsayısıyla orantılı olarak ölçeklendirir. Bireysel boyutları ölçeklendirmek model performansını artırabilirken EfficientNet tüm ağ özelliklerini optimal bir şekilde dengeleyerek mevcut kaynakları verimli bir şekilde kullanır ve genel ağ performansını artırır. Bu model ailesinin temelinde, MobileNet modellerindeki fikirlerden esinlenilen mobil ters şişkinlik evrişim (MBConv) bulunur. Bu, (depth-wise) ayrılabilir evrişimlerin kullanımını içerir. Noktasal ve depth-wise evrişim katmanlarını bir araya getirir. Ayrıca EfficientNet, MobileNet-V2'den ters çevrilen bağlantılar ve doğrusal darboğazlar gibi kavramları da içerir.

Veri büyütme

Aşırı öğrenme sorununu aşabilmek için veri kümesini yapay olarak genişletmemiz gerekmektedir. Mevcut veri kümesinin boyutunu veri büyütme kullanarak artırabiliriz. Burada temel amaç, bir fotoğraf veya video çekildiğinde görülen farkları, eğitim verisine mütevazı değişiklikler yaparak çoğaltmaktır.

Veri büyütme yöntemleri, eğitim verisini değiştirerek görüntü piksel temsilini korurken orijinal etiketi değiştirir. Bu kapsamda birçok yöntem (horizontal and vertical flips, Color hiccups, random crops, translations, and rotations vb.) veri büyütme için kullanılabilir. Bu yöntemler sıkça tercih edilir.

Mevcut eğitim değişikliklerinden sadece birkaçını uygulayarak eğitim örneklerimizi kolay bir şekilde ikiye katlayabilir ve genel sonuçlarımızı önemli ölçüde iyileştirebiliriz. Veri büyütme, orijinal verinin değiştirilmiş kopyalarını oluşturarak eğitim veri kümesinin boyutunu artırmak için kullanılan bir tekniktir.

Değerlendirme metrikleri

Genellikle bir makine öğrenimi modelinin yeni veriler üzerinde nasıl performans sergileyeceğini belirlemek için kullanılır. Dengeli veri kümelerinde doğruluk, hassasiyet ve geri çağırma gibi metrikler, sınıflandırma modellerini değerlendirmek için kullanışlı yollardır. Ancak veri dengesizse ROC/AUC gibi alternatif teknikler modelin performansını daha iyi bir şekilde değerlendirebilir. ROC eğrisi, sınıflandırıcının davranışı hakkında

ayrıntılı bilgi sunan bir eğridir ve tek bir sayıdan ibaret değildir. Ayrıca farklı ROC eğrilerini hızlı bir şekilde karşılaştırmak zaman alır.

Hassasiyet ölçütü, hedef sınıf dengeliyse yardımcı olur ancak dengesiz sınıflar için uygun seçenek olmayabilir. Eğitim verilerimizin %99'unun köpek resimlerinden oluştuğu ancak sadece %1'inin kedi resimleri olduğu bir durumu hayal edin. Köpek her zaman modelimiz tarafından tahmin edilir ve bize %99 oranında doğruluk sağlar. Veri sürekli olarak dengesizdir. Spam e-postaları, kredi kartı dolandırıcılığını ve yanlış tıbbi teşhisleri gösteren kanıtlarla desteklenir. Bu nedenle, daha iyi bir model değerlendirmesi yapılmak isteniyorsa kesinlik ve f1-skor gibi diğer metrikleri göz önünde bulundurmalıyız.

3. Deneysel çalışmalar

Eğitim sürecinde hiperparametre optimizasyonu için adam optimizer kullanılmıştır. Adam optimizer, temelde iki gradyan iniş yönteminin kombinasyonunu içerir:

Momentum: Bu yaklaşım, gradyan iniş algoritmasını gradyanların üssel ağırlıklı ortalamasını dikkate alarak hızlandırmak için kullanılır. Ortalamaların bu şekilde kullanılması, algoritmanın minimuma daha hızlı yaklaşmasını sağlar. Ayrıca bir öğrenme hızı zamanlayıcı mekanizması kullanılır. Öğrenme hızı zamanlamaları ile önceden tanımlanmış bir takvime göre öğrenme hızının ayarlanması amaçlanır.

Bir referans modeli: Bu yöntem, genetik algoritmanın gelişimi için baştan tasarlanmıştır. Ayrıca derin öğrenme modelleri olarak ResNet18, VGGNet16, VGGNet19 ve EfficientNet kullanılır. Bu üç CNN mimarisi, ImageNet veri kümesinde önceden eğitilmiş transfer öğrenme ile eğitilmektedir.

Transfer öğrenme, önceden edinilmiş bilgi ve becerilerin yeni öğrenme veya problem çözme durumlarında kullanılması anlamına gelir. Böylece önceki ve gerçek öğrenme içeriği ile süreçler arasındaki benzerlikler ve benzetmeler önemli bir rol oynayabilir.
Bu modeller, FER2013 ve RAF-DB veri kümeleri için yedi temel duygu üzerinde eğitilebilecek şekilde özelleştirilmiştir. Her modele uygun eğitim aşamasında özel katmanlar gösterilerek sonuçlar tartışılmıştır.
Aşağıda, her modelin eğitim ve doğrulama uygunluğu, çapraz entropi kaybını gösterir. Bunun devamında tüm bu modeller test kümesinde değerlendirilir. FER2013 veri kümesinde ResNet18, VGGNet16, VGGNet19 ve EfficientNet kullanılarak elde edilen sonuçlar, bu yaklaşımları yalnızca RAF-DB veri kümesinde değerlendirme imkanı da sağlar.

Veri kümesi analizi

FER2013 veri kümesi, 48x48 piksel boyutundaki yüzlerin gri tonlamalı görüntülerinden oluşur. Yüzler, her birinin yaklaşık olarak aynı konumda olduğu ve benzer bir alanı kapladığı şekilde otomatik olarak kaydedilmiştir. Her yüz, yedi kategoriye atanmıştır. Bu kategoriler; 0 öfke, 1 iğrenme, 2 korku, 3 mutluluk, 4 üzüntü, 5 sürpriz ve 6 normal (nötr) duygu şeklindedir. Eğitim verisinin test kısmında 7178 örnek bulunurken; eğitim kısmında 28,709 örnek bulunur.

RAF-DB veri kümesi, Real-world Affective Faces Database (RAFDB) içinde bulunan basit veya karmaşık ifade etiketleri yaklaşık olarak 30,000 yüz resminden oluşur. Sadece temel duygularla işaretlenen 12,271 örnek eğitim bölümünde kullanılmıştır.

Veri kümesi yedi etiket içerir. İğrenme etiketi, FER2013 veri kümesinde diğer etiketlere göre daha az görüntü içerir ve öğrenme aşamasında zorlayıcı olabilir. RAF-DB'de korku etiketi, diğer etiketlere göre daha az görüntü içerirken iki veri kümesinde de mutlu etiketinin zengin bir görüntü yoğunluğu bulunmaktadır.

Temel CNN modeli

Genetik algoritmanın sonraki gelişim sürecinde başvuru kaynağı olması amacıyla sıfırdan temel bir model geliştirilir. Bu bölümde genetik algoritma ve derin öğrenme modelleri olan ResNet18, VGGNet16 ve EfficientNet kullanılır. Bu üç CNN mimarisi, ImageNet veri kümesinde önceden eğitilmiş olarak transfer öğrenme ile eğitilir. Derin sinir ağları daha zor eğitilir. Bu nedenle eğitimi kolaylaştırmak için bir kalıntı öğrenme çerçevesi kullanılır. Ağ derinliği arttıkça doğruluk yoğunlaşır ve ardından hızla düşer. Daha fazla katman eklenmesi, doğruluğu her zaman artırmak anlamına gelmez ancak kalıntı (*residual*) çerçevesi daha derin katmanların kolay eğitilmesini sağlar. Bu kalıntı bloğu, mimarimiz için temel bir yapı taşı olarak kullanılır. Temel model, 0.001 öğrenme oranı ile 300 epok boyunca eğitilmiştir. Bu hiper parametre ayarı, farklı parametreler arasında yapılan zorlu bir araştırma sürecinde kullanılmıştır. Model, doğrulama kümesinde %60,05 oranında test doğruluğu elde etmiştir. Bu sonuç, 48x48 piksel boyutunda gri tonlamalı görüntülerle uğraştığımız veriye göre iyi bir doğruluk çıktısı gösterir.

Genetik algoritma kullanılarak uyarlanan evrişimli sinir ağları mimarisi

Bu bölümde, önerilen algoritmanın temel adımları ve çerçevesi sunulduktan sonra süreç özetlenmektedir. Önerilen algoritma, özel olarak girdi CNN mimarisi, popülasyon ve yavru boyutları ile nesil sayısını sağlayarak işlemeye başlar. Bir dizi evrimsel süreç aracılığıyla girdi CNN'leri için geliştirilmiş bir tasarım keşfeder. İlk olarak girdi CNN mimarisi başlangıç popülasyonunu hesaplamak için önemli ölçüde mutasyona uğrar. Evrim, ardışık nesil sayısı için çalışmaya başlar. Yeni bir yavru hesaplanır. Sağlanan veri kümesi kullanılarak değerlendirilir ve evrimin her aşamasında önceki adımda üretilen popülasyonla karşılaştırılır. Daha ayrıntılı olarak ifade edilirse yeni çocuğun üretilmesi, popülasyondan iki birey seçilmesi, genetik operatörleri kullanarak birleştirmeyi ve belirli bir CNN mimarisini kodlayan bir kişi üretmeyi içerir. Yeni birey daha sonrasında değerlendirilir ve popülasyonla karşılaştırılır. Mevcut popülasyon, bir sonraki nesilde yaşayacak birey sayısını hesaplamak için kullanılır. Önerilen teknik, rastgele bir aramanın nasıl akıllıca kullanılabileceğini göstermektedir. Ayrıca önerilen algoritma rastgele olmasına rağmen aslında tamamen rastgele değildir. Bunun yerine, daha iyi performans gösteren arama alanlarını odaklamak için önceden verilerden yararlanır. CNN mimarisinin katmanlarını birkaç nesil boyunca seçer. Rastgele keşif yoluyla bilgi kazanır ve daha iyi modelleri seçmek için bildiklerini kullanmaya başlar. Farklı mimarileri karşılaştırmak ve ardından en iyi mimariyi seçmek için test doğruluğunu kullanır. Prosedür, tamamen eğitilmiş ve uygun bir CNN modeli üretene kadar birçok nesil boyunca devam eder.

Önerilen algoritmanın başarısı, parametre seçimine bağlıdır. Parametreler uygun şekilde ayarlanmışsa çıktı, CNN'nin sınıflandırma hatası, girdi CNN'ninkinden belirgin şekilde daha düşüktür. Çalışmada genetik algoritma uygulanarak FER2013 veri kümesine uygun en iyi model bulunur. Algoritma; 8 nesil (G), 2 yavru (O) ve popülasyon boyutu (P = 4) ile çalıştırılır. Genetik algoritmadan elde edilen sonuçlara göre popülasyonun en iyi modeli Net1 adlı modeldir ve 372,423 parametreye sahiptir. En iyi modeli elde ettikten sonra adam optimizer kullanılarak 300 epok boyunca eğitim sürdürülür. Genetik algoritma modeli

sadece beş epok boyunca eğitilir ve her bireyin kaybı, modelin uygunluk fonksiyonu referans alınarak karşılaştırılır. Bu nedenle, modeli daha fazla epok boyunca eğiterek modelin ulaşabileceği en iyi doğruluğa ulaşılması istenmektedir. Eğitim süresince doğruluk eğrisine göre modelin %62,598 oranında doğruluk elde ettiği ve doğrulama kaybının ise 0,9985 olduğu belirtilmiştir.

EfficientNet model

Bu model, FER2013 ve RAF-DB veri kümeleri üzerinde eğitilmiştir. Modeli özelleştirmek için ImageNet veri kümesinin 1000 sınıfını temsil eden 1000 nöron içeren üst katmanı, sadece yedi temel duyguyu temsil eden nöronlarla değiştirilir. Başlangıç öğrenme hızı 0.005 olarak ayarlanır. Scheduler learning rate kullanılır ve doğrulama uygunluğu son beş epok boyunca artmazsa öğrenme hızı 0,3 faktörüyle azaltılır. Eğitim süreci 40 epok sonra durdurulur. Çünkü model, eğitim verisi ile ilgili aşırı öğrenme sağlamaya ve doğrulama uygunluğunu azalmaya başlar. Model, en iyi doğrulama için %67,09 değerini elde eder ve ilk epok sonrasında doğrulama kümesinde %49 ile eğitime başlar (transfer öğrenme nedeniyle). RAF-DB veri kümesinde de eğitim işlemi 40 epok sonra durdurulur. Çünkü model eğitim verisine fazlasıyla uyum sağlamaya başlar. Bu veri setinde model, %84,547 doğrulama oranına ulaşır.

ResNet18 modelini eğitme

Transfer öğrenme, ImageNet veri kümesinde önceden eğitilmiş ResNet18'i daha iyi eğitmek için kullanılır. Modeli seçilen iki veri kümesinde eğitiyoruz. Veri boyutumuza uygun ResNet18 sürümünü seçiyoruz. Bu sürüm, eğitim verisine uyum sağlayabilecek kapasitede olduğu için modelin karmaşıklığı sorunumuza uygundur. Model, 1000 nöron içeren üst katmanı, yalnızca 7 nöron içeren bir katmanla özelleştirilir. Bu, yedi temel duygu sınıfımıza uygun bir yapıdır. Her iki veri kümesi için de 0,0005 öğrenme hızı ile başlıyoruz. Öğrenmeyi doygunluğa ulaşana kadar azaltmak için bir zamanlayıcı öğrenme hızı kullanıyoruz. FER2013 veri kümesinde eğitim doğruluğu ve çapraz entropi kaybı var. Model, eğitim verisine aşırı öğrenme sağlamaya başladığı için 40 epok sonrasında eğitim durdurulmaktadır. Ayrıca doğrulama oranı azalmaya başlamıştır. En iyi doğrulama oranı %68,07 olarak elde edilmiştir. RAF-DB veri kümesinde de eğitim 40 epok sonrasında modelin aşırı öğrenme sorunu sebebiyle durduruldu. Bu veri kümesinde en iyi doğrulama oranı ise %86,02 olarak elde edilmiştir.

VGGNet16 modelini eğitme

VGGNet16 modeli, iki veri kümesi üzerinde eğitilirken bu modele fazlaca özelleştirme katmanı eklenmiştir. İlk olarak son ortalama havuzlama katmanının çıktı boyutunu (7, 7) yerine (1, 1) olarak değiştirilmiştir. Bu değişim, global ortalama havuzlama kullanılacağı anlamına gelir. Ayrıca sadece yedi çıkış nöronuna sahip üst katman değiştirilerek yedi sınıf temsil edilmiştir. Bu özelleştirme ise 138M parametreyi aşan sayıyı 33.6M parametreye düşürmeye yol açmıştır. Adam optimizerin, başlangıç öğrenme hızı 0.0005 olarak belirlenmiş ve diğer algoritmalarda olduğu gibi veri artırma modülünün parametrelerini ayarlamak için veri artırma boru hattı kullanılmıştır. En iyi düzenleme sonucunu veren ve eğitim verisinde aşırı öğrenme etkisini azaltan en iyi boru hattını elde etmek için farklı boru hatlarında kapsamlı bir arama yapılır. Model, 40 epok boyunca FER2013 veri kümesinde %70,16 doğrulama oranı elde etmiş ve ardından eğitim durdurulmuştur. Çünkü model eğitim verisi üzerinde aşırı öğrenme sorunuyla karşılaşmaya başlamıştır. RAF-DB veri kümesinde

de eğitimler 40 epok sonrasında durdurulmuştur. Burada da aşırı öğrenme sorunu ortaya çıkmıştır. Bu veri kümesinde en iyi doğrulama oranı %85,7 olarak elde edilmiştir.

VGGNet19 modelini eğitme

VGGNet19 modelini eğitirken, bu modelde daha fazla özelleştirme katmanı eklenmiştir. İlk olarak son ortalama havuzlama katmanının çıktı boyutu (7, 7) yerine (1, 1) olarak değiştirilmiştir. Bu değişiklik, global ortalama havuzlamanın kullanacağı anlamına gelir. Ayrıca üst katmanı sadece 7 çıkış nörona sahip olacak şekilde değiştirilmiş ve bu da 7 sınıfımı temsil etmiştir. Buradaki özelleştirme ise 138M parametreyi aşan sayıyı, 45.2M parametreye düşürmeyi sağlamıştır. Adam optimizerin başlangıç öğrenme hızı 0.0005 olarak belirlenmiş ve düşük seviyeli öğrenme hızı zamanlayıcısı kullanılmıştır. Ayrıca yeni veri artırma boru hattı kullanılır. Veri artırma modülünün parametrelerini ayarlamak için farklı boru hatlarında kapsamlı bir arama yapılır ve en iyi düzenleme sonucunu veren boru hattı bulmaya çalışılır. Bu durum, eğitim verisinde aşırı öğrenme etkisini azaltır. VGGNet19 özel modeli, 100 epok boyunca FER2013 veri kümesinde %71 doğrulama oranı elde edilmiştir. Bu işlemin ardından eğitim işlemi durdurulur. Çünkü model eğitim verisinde aşırı öğrenme sorunuyla karşılaşmıştır. RAF-DB veri kümesinde VGGNet modelinin eğitim ve doğrulama değeri ile çapraz entropi kaybı, 100 epok boyunca eğitim süresine göre belirlenir. Model, en iyi doğrulama oranı için %85,87 değerini elde etmiştir. Her iki veri kümesi için de aynı optimizer ve öğrenme hızı zamanlayıcı parametresi ile aynı veri artırma yapısı kullanılmıştır.

4. Sonuçlar ve tartışma

Bu çalışmada kullanılan her modelin performansı ayrı ayrı değerlendirilmiş ve modellerin test verileri üzerinde nasıl bir performans sergilediği hakkında daha fazla bilgi sunulmuştur. Modelin verimliliği, farklı metrikler ile ölçülmüş ve her etiket için hassasiyet, kesinlik ve f1-skor değerleri hesaplanmıştır. İlk olarak karmaşıklık matrisi ile hesaplanmış ve karmaşıklık matrisi, modelin yaptığı hata türlerini ve bunların farklı etiketler arasında nasıl dağıldığını göstermiştir. Bu işlem sınıflandırma algoritmasının performansını özetlemek için kullanılan bir tekniktir. Sınıflandırmanın doğruluğu yalnızca her sınıfta eşit sayıda gözlem olduğunda veya veri kümesinde iki sınıftan fazlası olduğunda yanıltıcı olabilir. Bir karmaşıklık matrisini hesaplamak, sınıflandırma modelimizin neyi doğru yaptığını ve hangi tür hatalar gerçekleştirdiğini daha iyi anlamamıza yardımcı olabilir.

Hassasiyet, kesinlik ve f1-skor değerleri, sadece genel doğruluk değerine bakmak yerine bir sınıflandırıcının ne kadar iyi sınıflandırma yaptığına dair iyi fikirler verir. Hassasiyet, yapılan pozitif tahminlerin kaçının doğru olduğunu (gerçek pozitifler) ölçer. Kesinlik, sınıflandırıcının doğru tahmin ettiği pozitif vakaların tüm pozitif vakalar içindeki sayısını ölçer. Bazen duyarlılık olarak da adlandırılır. F1-skor ölçütü ise hassasiyet ve kesinlik ölçütlerinin birleşimini sunar. Genellikle hassasiyet ve kesinlik harmonik ortalama ile hesaplanır. Harmonik ortalama, değerlerin bir "ortalama" hesaplamasının başka bir yoludur ancak daha düşük değerlere karşı daha bu ölçüt daha duyarlıdır ve doğru bir çıktı verir.

Temel model

Temel model, gelecekteki çalışma yönünü belirler ve sonraki derin öğrenme modellerinde transfer öğrenmenin etkisini incelemek için baştan tasarlanır.Ayrıca temel modelin hassasiyeti, kesinliği ve f1-skoru, her etiket üzerinden bağımsız olarak hesaplanır. Bu, modelin görünmeyen test verilerinde nasıl bir performans gösterdiği ve etiketlerin en çok

nerede hata yaptığı hakkında daha fazla sezgi elde etmemizi sağlar.

Tüm bunlar, problem karmaşıklığını anlamamıza ve veri kümesini daha verimli bir şekilde hafızaya alabilen karmaşık modellerle test etmek için transfer öğrenmeyi kullanmanın önemli olduğunu anlamamıza olanak tanır. F1-skor sonucu olarak %60,54 değeri elde edilmiştir.

Genetik algoritma

Bu algoritma, geliştirilen modele göre daha iyi bir model sunabilmek amacıyla ilgili probleme uyarlanmıştır. Genetik algoritma, doğruluğu %1.8 artırırken; model parametre sayısını 1,672,775'ten 372,423'e düşürmüştür. Bu model, ilk modele göre 5 kat daha küçüktür. Model verimliliği çeşitli metrikler kullanılarak kısaca ele alınmıştır. Testlerde f1-skor sonucu %62.598 oranında bulunmuştur.

EfficientNet model

EfficientNet modelinin, iki veri kümesi olan FER2013 ve RAF-DB üzerinde nasıl performans gösterdiği konusunda daha fazla bilgi sağlamak için karışıklık matrisi, normalize edilmiştir. Bu matris; hassasiyet, kesinlik ve f1-skor için hesaplanmıştır. Normalize edilmiş karmaşıklık matrisi, her etiket için görüntü sayısına bölünmüş karışıklık matrisidir ve hata dağılımını hesaplamak için kullanılır.

FER2013 veri kümesindeki etkili modelin karmaşıklık matrisi ve normalize edilmiş karışıklık matrisinde mutlu ve şaşkın etiketlerine bakıldığında mutlu için %84 doğruluk ve şaşkın için %79 doğruluk oranı en iyi sonuçları verir. Model, en fazla hata ile karşılaşılan korku, öfke, normal ve üzgün etiketleri konusunda kısıtlara sahiptir.

Bu yedi temel duygu için hassasiyet, geri çağırma ve f1-skor hesaplanmıştır. Ayrıca hangi görüntülerin bu sonuca katkıda bulunduğunun gösterilmesi, EfficientNet modelinin FER2013 veri kümesinde nasıl performans gösterdiği hakkında daha fazla bilgi sağlar. Şaşkın sınıfı, normal ve üzgün gibi daha az görüntüye sahip olmasına rağmen 0.78 f1-skorunu alır.
RAF-DB test kümesinin etkili net modelinin karmaşıklık matrisi ve normalize edilmiş karmaşıklık matrisinde RAF-DB, 100x100 boyutunda RGB görüntü verisine sahip olduğu için daha sağlam bir hata dağılımına sahiptir ve modelin daha fazla bilgi yakalamasına olanak tanır. RAF-DB modelinin sınıflandırma raporuna bakıldığında bu model, test kümesinde %84.547 doğruluk elde etmiştir. Model, mutluluk sınıfını daha doğru tahmin etmeyi sağlayan birçok örneğe sahip olduğu için mutlu görüntülerini %0.93 f1-skoru ile doğru tahmin edebilir. Ayrıca şaşkın, üzgün ve normal etiketleri %0.8'in üzerinde f1-skoru sağlamıştır.

ResNet18 model

ResNet18 modeli, önerilen yaklaşımın test kümesinde nasıl performans gösterdiğine dair daha fazla bilgi sağlamak için değerlendirilmiştir. Her veri kümesinde hem karmaşıklık matrisi hem de normalize edilmiş karmaşıklık matrisi hesaplanmış ve sonuçlar sınıflandırma raporunda sunulmuştur. Bu rapor, yedi temel duygu etiketi için hassasiyet, kesinlik ve f1-skor çıktılarını gösterir.

FER2013 veri kümesindeki modelin sınıflandırma raporu, modelin test kümesinde toplam %68,07 oranında doğruluk elde ettiğini göstermektedir.
RAF-DB test kümesindeki ResNet18 modelinin karmaşıklık matrisi ve normalize edilmiş karmaşıklık matrisinden hareketle modelin bu veri kümesinde daha sağlam olduğunu ve literatürdeki diğer çalışmalara kıyasla makul bir doğruluk elde edebildiğini kolayca gözlemleyebiliriz. RAF-DB test kümesinde model, %86,02 test doğruluğu elde etmiştir. Model, mutlu etiketinde %94 f1-skoruna ulaşırken; şaşkın, üzgün, öfkeli ve nötr etiketlerinde de 0.8'in üzerinde f1-skorunu elde edilmiştir.

VGGNet16 model

VGGNet16 modeli, FER2013 veri kümesinde %70,2 test doğruluğu elde etmişt**ir**. VGGNet'in orijinal modeli 138M parametreye sahiptir. Bu sayı diğer modellerle karşılaştırıldığında çok büyüktür ve karmaşıklığı çok yüksektir. Ancak eklediğimiz özelleştirmeler parametre sayısını 33.6M parametreye düşürür ve bu daha makul bir değer olarak kabul edilir.

Tüm etiketlerin hassasiyeti, kesinlik ve f1-skoru, bu metrikleri hesaplamada her etiketin kaç görüntüye katkıda bulunduğunu göstermiştir. Tablodan, her etiketin f1-skorununarttığı ve test kümesindeki genel doğruluğun %70,2 olduğu görülebilir. ResNet18 modeli bu veri kümesinde en iyi sonucu elde etti ancak VGGNet modeli test kümesinde %85,88 elde ederken ResNet18 %86.02 elde eder. Doğruluk farkı büyük değildir fakat VGGNet modelinin daha fazla parametreye sahip olduğunu ve karmaşıklığının çok yüksek olduğunu unutmamak gerekir.

VGGNet19 model

VGGNet19 modeli, test kümesinde değerlendirilmiştir. Bu model, FER2013 veri kümesinde en iyi sonucu elde ederek test doğruluğu olarak %71,2 oranına ulaşmıştır. VGGNet'in orijinal modeli 138M parametreye sahiptir. Bu sayı diğer modellerle karşılaştırıldığında çok büyüktür ve karmaşıklığı çok yüksektir. Ancak eklenen özelleştirmeler parametre sayısını 40,2M parametreye düşürür ve bu daha makul bir değerdir. Tüm etiketlerin hassasiyeti, kesinliği ve f1-skor metrikleri ve bu metrikleri hesaplamada her etiketin kaç görüntüye katkıda bulunduğu vurgulanmıştır. Her etiket için f1-skorunun arttığı ve test setindeki genel doğruluğun %71.02 olduğu gözlemlenmiştir.

5. Sonuç

Yüz ifadeleri; insan duygularını, düşüncelerini ve niyetlerini daha iyi anlamak için etkin bir şekilde kullanılabilir. Çünkü çok miktarda sözle ifade edilmeyen ve etiketsiz bilgiler sağlanmaktadır. Bu çalışmada belirtilen stratejinin, yüz ifadesi ile ilgili tanıma performansında başarılı olduğu ve daha önce literatürde ele alınmış konulara yeni bir bakış açısıyla katkıda bulunduğu gözlemlenmiştir. Kullanılan yeni yöntem, yüz ifadelerini daha doğru ve etkili bir şekilde tanımayı ve sınıflandırmayı mümkün kılarak hesaplama maliyetlerini ve çalışma süresini azaltırken başarım oranlarını arttırmıştır. Bu yeni model, yüz görüntülerinin kategorizasyon doğruluğunu artırmak amacıyla oluşturulmuştur. Sonuçlar, derin öğrenme destekli yüz ifadesi tanıma ve sınıflandırma yöntemlerinin, gelişmiş yüz tanıma, doğruluk, özellik ve ifade yorumu aracılığıyla üretkenliği artırdığını göstermektedir.

Yüz ifadesi duygu tanımlama, güvenlik, sağlık ve insan-makine etkileşimi de dahil olmak üzere birçok alanda uygulamaları olan ilginç bir araştırma konusudur. Bu alandaki araştırmacılar, yüz ifadelerini okuma, kodlama ve çıkarma teknikleri oluşturarak bilgisayar tahminlerini geliştirmeye çalışmaktadır. Derin öğrenmenin olağanüstü başarısı, performans artışı için birçok farklı mimarinin kullanılmasına katkı sağlamıştır.

Bu çalışmada CNN yaklaşımı, FER2013 veri kümesi üzerinde kullanılmış ve artıklı öğrenme bloklarının etkisini modelin performansını artırmak için incelemiştir. Katman ve nöron sayısı, farklı veri kümeleri için CNN uygulamasının sezgisine dayanarak belirlenmiştir. Öğrenme hızı ve zamanlayıcı gibi etkenleri seçmek için deneme-yanılma yaklaşımı kullanılmış ve uygun parametre değerleri belirlenmiştir.

Bir başka teknik, genetik algoritma kullanılarak CNN mimarisinin evrimleştirilmesini ele alır. Burada görüntü sınıflandırma problemini çözebilmek için kullanılan evrişimli sinir ağı, genetik algoritma kullanılarak iyileştirilmiştir. Genetik algoritma, ilgili veri çerçevesine uyarlanarak temel modelden daha uygun bir model elde etmek amacıyla kullanılmıştır. Burada doğruluğu %1,8 oranında artırıp model boyutunu temel modele göre beş kat azaltmak mümkün olmuştur.

Önerilen model, FER2013 veri seti kullanılarak yüz tanıma sistemlerinin doğruluğunu artırmak için diğer modellerle birleştirilebilir. Çünkü bu işlemlerde hesaplama maliyeti daha düşüktür. FER2013, her sınıf için az sayıda örnek içeren karmaşık bir veri kümesidir. Ancak doğruluğu artırmak amacıyla her sınıftaki örnek sayısı uygun miktarda artırılabilir. Bu durumu gerçekleştirmek için transfer öğrenme teknikleri kullanılarak VGGNet16, VGGNet19, ResNet18 ve EfficientNet-B0 modelleri FER2013 ve RAF-DB isimli iki veri kümesinde yeniden eğitilmiştir. Önceden eğitilmiş CNN mimarileri, bu veri kümelerinde büyük bir performans sergilemiştir.

Kaynaklar

1. Mnih, V., and Hinton, G. E. (2010). *Learning to detect roads in high-resolution aerial images*. In Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September, Proceedings, Part VI 11. Springer Berlin Heidelberg, pp. 210-223
2. Mnih, V., and Hinton, G. E. (2012). *Learning to label aerial images from noisy data*. In Proceedings of the 29th International conference on machine learning (ICML-12), pp. 567-574.
3. Zhang, Z., Wang, Y., Liu, Q., Li, L., and Wang, P. (2016, July). *A CNN based functional zone classification method for aerial images*. In 2016 IEEE international geoscience and remote sensing symposium (IGARSS). IEEE pp. 5449-5452.
4. Rosenstein, D., and Oster, H. (1988). *Differential facial responses to four basic tastes in newborns*. Child development, 1555-1568.
5. Internet: *FER2013 dataset*, url: https://www.kaggle.com/datasets/msambare/fer2013, access date: December 1, 2023. Last accessed date 12/12/2023.
6. Goodfellow, I. J., Erhan, D., Carrier, P. L., Courville, A., Mirza, M., Hamner, B., and Bengio, Y. (2013). *Challenges in representation learning: A report on three machine learning contests*. In Neural Information Processing: 20th International Conference, ICONIP 2013, Daegu, Korea. Proceedings, Part III 20. Springer berlin Heidelberg,pp. 117-124.

7. Internet: *RAF-DB*, url: https://www.kaggle.com/datasets/shuvoalok/raf-db-dataset. Last access date: 01/12/2023.
8. 8.He, K., Zhang, X., Ren, S., and Sun, J. (2016). *Deep residual learning for image recognition*. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778.
9. Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv*:1409.1556.
10. Tan, M., and Le, Q. (2019, May). EfficientNet: *Rethinking model scaling for convolutional neural networks*. In International conference on machine learning. PMLR, pp. 6105-6114.
11. Pashine, S., Dixit, R., and Kushwah, R. (2021). Handwritten digit recognition using machine and deep learning algorithms. *arXiv preprint arXiv*, 2106.12614.
12. Kollias, D., Tzirakis, P., Nicolaou, M. A., Papaioannou, A., Zhao, G., Schuller, B., and Zafeiriou, S. (2019). Deep affect prediction in-the-wild: Aff-wild database and challenge, deep architectures, and beyond. *International Journal of Computer Vision*, *127*(6-7), 907-929.

Bilim Kodu	:	92432
Anahtar Kelimeler	:	Bilgi teknolojileri, duygu tanıma, derin öğrenme, genetik algoritma, VGGNet, ResNet, EfficientNet
Sayfa Adedi	:	85
Danışman	:	Dr. Öğr. Üyesi Yılmaz ATAY

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation and gratitude to the following individuals and organizations who have played a significant role in the completion of this thesis:

Thesis Advisor, Dr. Yılmaz ATAY: I am profoundly thankful for the guidance, wisdom, and encouragement provided throughout the research process. Your expertise and support have been invaluable, shaping both the content and direction of this work.

Family: To my Wife, your unwavering support has been my anchor. Your belief in me and your encouragement during challenging times have been indispensable. This achievement is as much yours as it is mine.

Friends: I am grateful to my friends for their understanding, encouragement, and occasional distractions during this intense period of study. Your camaraderie provided balance and perspective, reminding me of the importance of life beyond academia.

Institutional Support: I would like to acknowledge the Gazi University Graduate School of Natural and Applied Sciences for providing the necessary resources and facilities essential for the successful completion of this research. The academic environment has been conducive to intellectual growth and exploration.

Funding Agencies: This research was made possible through the generous support of the Iraqi Federal Board of Supreme Audit. Their financial assistance played a crucial role in the execution of this project.

CONTENTS

Page

ABSTRACT IV

ÖZET v

ACKNOWLEDGEMENTS xix

CONTENTS xx

LIST OF TABLES xxii

LIST OF FIGURES xxiii

1. INTRODUCTION 1

2. LITERATURE REVIEW 7

2.1. Facial Expression Databases 8

2.2. Deep Facial Expression Recognition 10

3. METHODOLOGY & PROPOSED APPROACH 15

3.1. Convolutional Neural Network & Deep Learning Approaches 16

3.1.1. Genetic algorithm *19*

3.1.2. VGGNet *20*

3.1.3. ResNet *22*

3.1.4. EfficientNet *24*

3.2. Data Augmentation 27

3.3. Evaluation Metrics 30

3.3.1. Confusion matrix *31*

3.3.2. Receiver Operating Characteristic (ROC) Curve 32

3.3.3. Accuracy 32

3.3.4. Recall 33

3.3.5. Precision 33

3.3.6. F1 score 34

4. EXPERIMENTAL STUDIES 35

Page

4.1. Dataset Analysis 36
4.2. Build Base Line CNN Model 38
4.2.1. Training the base line model *42*
4.3. Evolving Architecture for CNNs Using Genetic Algorithm 45
4.3.1. Algorithm overview *46*
4.3.2. Training genetic algorithm designed models *47*
4.4. Training EfficientNet Model 50
4.5. Training ResNet18 Model 52
4.6. Training VGGNet16 Model 55
4.7. Training VGGNet19 Model 58
4.8. Training Summary 62
5. RESULTS AND DISCUSSION 63
5.1. Base Line Model 63
5.2. Genetic Algorithm 65
5.3. EfficientNet Model 66
5.4. ResNet18 68
5.5. VGGNet16 Model 70
5.6. VGGNet19 72
5.7. The Comparison of Models 74
5.8. Study Robustness 75
5.9. Testing Summary 76
6. CONCLUSION AND FUTURE DIRECTIONS 77
6.1. Outline of the Contribution 77
6.2. Overall Conclusion 77
REFERENCES 79
CURRICULUM VITAE 85

LIST OF TABLES

Table **Page**

Table 1.1. The six basic emotions 3

Table 2.1. Gesture recognition databases 9

Table 2.2. State-of-the-art results on FER2013 data set. 11

Table 2.3. State-of-the-art results on RAF-DB data set. 12

Table 2.4. Concordance Correlation Coefficient (CCC) 13

Table 2.5. Mean Squared Error (MSE) 14

Table 4.1. Genetic Algorithm final population results. 48

Table 5.1. The classification report of the baseline model on the FER2013 data set 64

Table 5.2. The classification report of the designed genetic algorithm model of the FER2013 data set. 66

Table 5.3. The classification report of the EfficientNet-B0 model on the FER2013 67

Table 5.4. The classification report of the EfficientNet-B0 model on the RAF-DB 68

Table 5.5. Classification report of the ResNet18 model on the FER2013 69

Table 5.6. Classification report of the ResNet18 model on the RAF-DB 70

Table 5.7. The classification report of the VGGNet16 model on the FER2013 71

Table 5.8. The classification report of the VGGNet16 model on the RAF-DB 72

Table 5.9. The classification report of the VGGNet19 model on the FER2013 73

Table 5.10. The classification report of the VGGNet19 model on the RAF-DB 74

Table 5.11. All Models Results on FER2013 Test set. 75

Table 5.12. All model results on RAF-DB Test set. 75

Tablet 5.13. Comparison of the results on the FER2013. 76

Tablet 5.14. Comparison of the results on the RAF-DB 76

LIST OF FIGURES

Figures **Page**

Figure 1.1. Thesis organization 6

Figure 2.1. Annotations for valence and arousal over a section of a video, with the associated frames. 10

Figure 3.1. Pipeline Framework 16

Figure 3.2. Convolutional neural network (CNN) architecture 18

Figure 3.3. The flowchart of the genetic algorithm 20

Figure 3.4. Overview of VGGNet's (a) VGGNet16 (b) VGGNet19 architectures 22

Figure 3.5. Overview of ResNet-18 architecture 24

Figure 3.6. The new Swish activation function 25

Figure 3.7. Overview of the EfficientNet-B0 model 26

Figure 3.8. Illustration of the squeeze-and-excitation concept 26

Figure 3.10. FER2013 data set before and after applying our augmentation. 30

Figure 3.11. RAF-DB data set before and after applying our augmentation..... 30

Figure 3.12. Confusion Matrix for Binary Classification 32

Figure 4.1. Bar graph illustrating the Kaggle's fer2013 data set..... 37

Figure 4.2. Bar graph illustrating the RAF-DB data set..... 37

Figure 4.3. Some grayscale samples from the FER2013 data set..... 38

Figure 4.4. Some RGB image samples from RAF-DB data set. 38

Figure 4.5. Residual learning: a building block..... 39

Figure 4.6. Baseline model architecture 41

Figure 4.7. Training accuracy on the training set..... 42

Figure 4.8. The loss of both training and validation set.. 43

Figure 4.9. The AUC metric of the training and validation set..... 43

Figure 4.10. Reduce on Plateau Learning rate scheduler.. 44

Figure 4.11. The flowchart of the proposed genetic algorithm 45

Figures **Page**

Figure 4.12. The proposed (GA) structure. 47

Figure 4.13. Optimal GA model Architecture. 48

Figure 4.14.Training accuracy over training 49

Figure 4.15. Training Cross Entropy Loss Curve. 49

Figure 4.16. Training accuracy versus 40 training epochs on the FER2013 data set. 50

Figure 4.17. Training Cross Entropy Loss Curve of FER2013 data set. 51

Figure 4.18. Training accuracy versus 40 training epochs on RAF-DB. 51

Figure 4.19. Training Cross Entropy Loss Curve on RAF-DB. 52

Figure 4.20. Training accuracy versus 40 training epochs on the FER2013 data set. 53

Figure 4.21. Training Cross Entropy Loss Curve of FER2013 data set. 53

Figure 4.22. Training accuracy versus 40 training epochs on RAF-DB. 54

Figure 4.23. Training Cross Entropy Loss Curve on RAF-DB. 54

Figure 4.24. Training accuracy versus 40 training epochs on the FER2013 data set. 56

Figure 4.25. Training Cross Entropy Loss Curve of FER2013 data set. 56

Figure 4.26. Training accuracy versus 40 training epochs on RAF-DB. 57

Figure 4.27. Training Cross Entropy Loss Curve on RAF-DB. 57

Figure 4.28. Training accuracy versus 100 training epochs on FER2013 data set. 60

Figure 4.29. Training Cross Entropy Loss Curve of FER2013 data set. 60

Figure 4.30. Training accuracy versus 100 training epochs on RAF-DB. 61

Figure 4.31. Training Cross Entropy Loss Curve on RAF-DB. 61

Figure 5.1. Confusion matrix result of the baseline model 64

Figure 5.2. Confusion matrix of the designed genetic algorithm model 65

Figure 5.3. The EfficientNet-B0 model on the FER2013 test set. 67

Figure 5.4. The EfficientNet-B0 model on the RAF-DB test set. 68

Figure 5.5. The ResNet18 model on the FER2013 test set 69

Figures **Page**

Figure 5.6. The ResNet18 model on the RAF-DB test set 70

Figure 5.7. The VGGNet16 model on the FER2013 test set. 71

Figure 5.8. The VGGNet16 model on the RAF-DB test set 72

Figure 5.9. The VGGNet19 model on the FER2013 test set. 73

Figure 5.10. The VGGNet19 model on the RAF-DB test set..... 74

1. INTRODUCTION

Deep learning has made significant strides in recent years, demonstrating breakthroughs in various computer vision domains, from scene identification to object detection. The implementation of techniques based on deep neural networks has significantly enhanced our capacity to interpret and understand data related to emotion recognition [1-3]. In this process, various deep learning architectures specifically designed for gesture recognition are utilized to develop a robust strategy suitable for overcoming the challenges of emotion recognition. Gestures, as a fundamental element of communication, play a pivotal role in computer vision applications [4]. This study progressively evolves from a basic strategy applied to straightforward data to a sophisticated approach tackling more intricate datasets. It has been focused on delving deeper into the complexity of emotions and exploring notable applications of computer vision in addressing complex emotional expressions in this study. Computer vision is a field of study that focuses on enabling computers to interpret and understand the visual world. This field involves developing up-to-date algorithms to analyse and interpret images and extract useful information. One of the most exciting research areas of computer vision is the emotion recognition problem. Emotion recognition focuses on the problem of identifying human emotions based on facial expressions, body language, and other non-verbal cues. This problem is a complex process that requires sophisticated algorithms and machine-learning techniques. The complexities of emotion recognition arise because emotions are often subtle and difficult to interpret. For example, a smile can indicate happiness but also show sarcasm or contempt. Similarly, a frown can show sadness, but it can also mean anger or frustration. To handle these complexities, computer vision researchers have developed a range of techniques that can analyse facial expressions and other non-verbal cues to identify emotions accurately. By doing so, it is possible to develop new techniques and algorithms to improve the accuracy and reliability of emotion recognition systems. This has important implications for a wide range of fields, including healthcare, education, and entertainment.

Problem definition

This thesis is about a process that affects effective communication, such as emotion prediction from facial expressions. This problem can be related to the following example. In Shakespeare's timeless play "Macbeth" (circa 1699), King Duncan faced betrayal by Thane

of Cawdor, leading him to the profound realization that "You cannot read the mind or gauge the gullibility of another person solely by their face." This quote can still apply today to some everyday situations such as communication misunderstandings. Yet, it is undeniable that humans excel in discerning emotions from facial expressions. This work aims to scrutinize the extent to which facial expressions reliably signal specific emotions and assess the efficacy of this mode of communication in gauging another person's emotional state. Nonverbal communication stands as a cornerstone of human interaction, with gestures emerging as a prominent form [4 - 6]. Gestures are broadly categorized into three types [7]:

- Intrinsic: Nodding, an example of an intrinsic gesture, appears to be an innate behaviour, utilized even by individuals blind from birth to convey affirmation or agreement [8].

- Extrinsic: An extrinsic gesture, such as turning to the side as a sign of refusal learned during early childhood, illustrates the adaptability of human communication. For instance, a baby may turn away from the mother's breast after having enough milk to signal fullness [9].

- Result of Natural Selection: Consider the widening of the nose to intake more oxygen, potentially indicating the body's preparation for combat or escape.

Computers are now integral to every aspect of human life. Understanding the human emotional state equips computers to adapt better, fostering improved cooperation. Gestures, a crucial form of communication, find applications in computer vision, specifically in gesture recognition. Body language encompasses diverse nonverbal indicators—facial expressions, body posture, gestures, eye movement, touch, and the use of personal space. An individual's inner state is expressed through various elements, including the positioning of hands and legs, as well as their sitting, standing, and movement styles.

Cultural differences significantly contribute to our challenge, given the strong culture-dependent nature of gestures [10,11]. While some gestures have globalized, representing a shared human experience, complex emotions may involve a combination of these universal gestures.

Fundamental disparities exist in how men and women communicate through body language,

with gender playing a substantial role. The same gesture may convey different emotions based on gender [5]. In general, men exhibit less facial expressiveness they smile less and display fewer emotions. This phenomenon arises from childhood teachings, where women are encouraged to adopt appeasement body language, fostering a disposition for cooperation and harmony [12]. Table 1.1. shows the general movement protocols for the six basic emotions to understanding these movement patterns provides valuable insights into how different emotions manifest in our physical behaviour. It's fascinating how our bodies express what words alone cannot convey.

Table 1.1. The six basic emotions [12 - 14]

Emotion	Associated Gestures
Fear	High heartbeat rate, legs and arms are crossing and moving, muscle tension: hands or arms clenched, elbows dragged inward, bouncy movements, legs wrapped around objects, breath held, conservative body posture and hyper-arousal body language.
Anger	Body spread, hands-on hips or waist, closed hands or clenched fists, palm-down posture, lift the right or left hand up, finger point with the right or left hand, finger or hand shaky and arms crossing.
Sadness	Body dropped, shrunk body, bowed shoulders, body shifted, trunk leaning forward, the face is covered with two hands, self-touch (disbelief), body parts covered or arms around the body or shoulders, body extended and hands over the head, hands kept lower than their normal positions, hands closed or moving slowly, two hands touching the head and moving slowly, one hand touching the neck, hands closed together and head bent.
Surprise	Abrupt backward movement, one hand or both moving toward the head, moving one hand up, both hands touch the head, one of the hands or both touching the face or mouth, both hands are over the head, one hand touching the face, self-touch or both hands covering the cheeks or mouth, head shaking and body shift or backing.
Happiness	Arms open, arms move, legs open, legs parallel, legs may be stretched apart, feet pointing to something or someone of interest and looking around, eye contact relaxed and lengthened.
Disgust	Backing, hands covering the neck, one hand on the mouth, one hand up, hands close to the body, body shifted, orientation changed or moving to a side and hands covering the head.

In addressing our problem, our objective is to systematically develop the system from a simple approach to a complex one, gradually transitioning from simple data to more

challenging datasets using an incremental learning approach. We adopt an incremental learning approach to delve deeper into the problem, seeking a comprehensive understanding of how computer vision can be effectively applied to intricate and demanding tasks.

Contribution

To establish a robust approach, we utilize two open-source datasets, namely FER2013 [15], [16] and RAF-DB[17]. These datasets enhance our insight into the Gesture recognition problem, and our primary contributions are summarised as follows:

- Constructing a baseline model from scratch as a reference for subsequent development. This initial model helps us comprehend the impact of transfer learning and the utilization of common deep-learning architectures.

- Employing various deep-learning models, including ResNet18[18], VGGNet16[19], VGGNet19[19], and EfficientNet-B0[20], contributes to achieving a robust model with reasonable accuracy on both datasets.

- Applying a Genetic Algorithm aids in designing a CNN model architecture from scratch. The fitness function for the population models is selected as the Cross-Entropy loss of the designed models.

- Evaluating the employed models on the test set, calculating the confusion matrix and normalized confusion matrix. Notably, on the FER2013 dataset, the VGGNet19 model achieves the best results, with a test accuracy of 71.02%. While the original VGGNet model with 138M parameters is excessively complex, our customization reduces the parameter count to 45.2M, offering a fairer comparison. The VGGNet19 attains an 85.85% test accuracy on the RAF-DB dataset, while the ResNet18 outperforms all other models with an 86.02% test accuracy.

These contributions underscore the significance of our work in advancing automated gesture recognition systems for complex tasks in computer vision.

Limitation

Automatic facial expression recognition systems can be made more dimensional by combining facial expression recognition with the idea of natural language processing (NLP). The future scope can play more significant roles in the e-health system and the provision of healthcare services if it is implemented.

Combining human language with facial expressions to predict the emotion of a human literally will be a significant improvement, providing more information to the system to predict the motion more accurately.

The proposed approach depends only on facial expression, and that is the main limitation of our work; we only tackle the problem from the Face emotion data sets FER2013 and RAF-DB and use the deep CNN architecture to gather information on human facial expressions.

Thesis organization

Firstly, the "Literature Review" section is presented. This section explores previous research on the Emotion Recognition problem, discussing the results and limitations of traditional approaches. It also briefly overviews different approaches and examines how deep learning and related architectures impact accuracy, influencing our research plan. The third section is the "Methodology & Proposed Approach." This section details each model architecture used and presents our pipeline design, describing each module within this pipeline. The fourth section is "Experimental Studies" discusses all employed experiments, trains the models, and illustrates the effect of different hyperparameters for data augmentation. Additionally, this section outlines each customization layer used for the deep learning architectures. The fifth section is the "Results and Discussion." It presents all obtained results and compares models based on their performance on the test sets for both datasets. The last section is the "Conclusion and Future Directions." This chapter highlights the main contributions of the study, and provides an overall conclusion. The basic contents of all sections in this thesis are given in Figure 1.1.

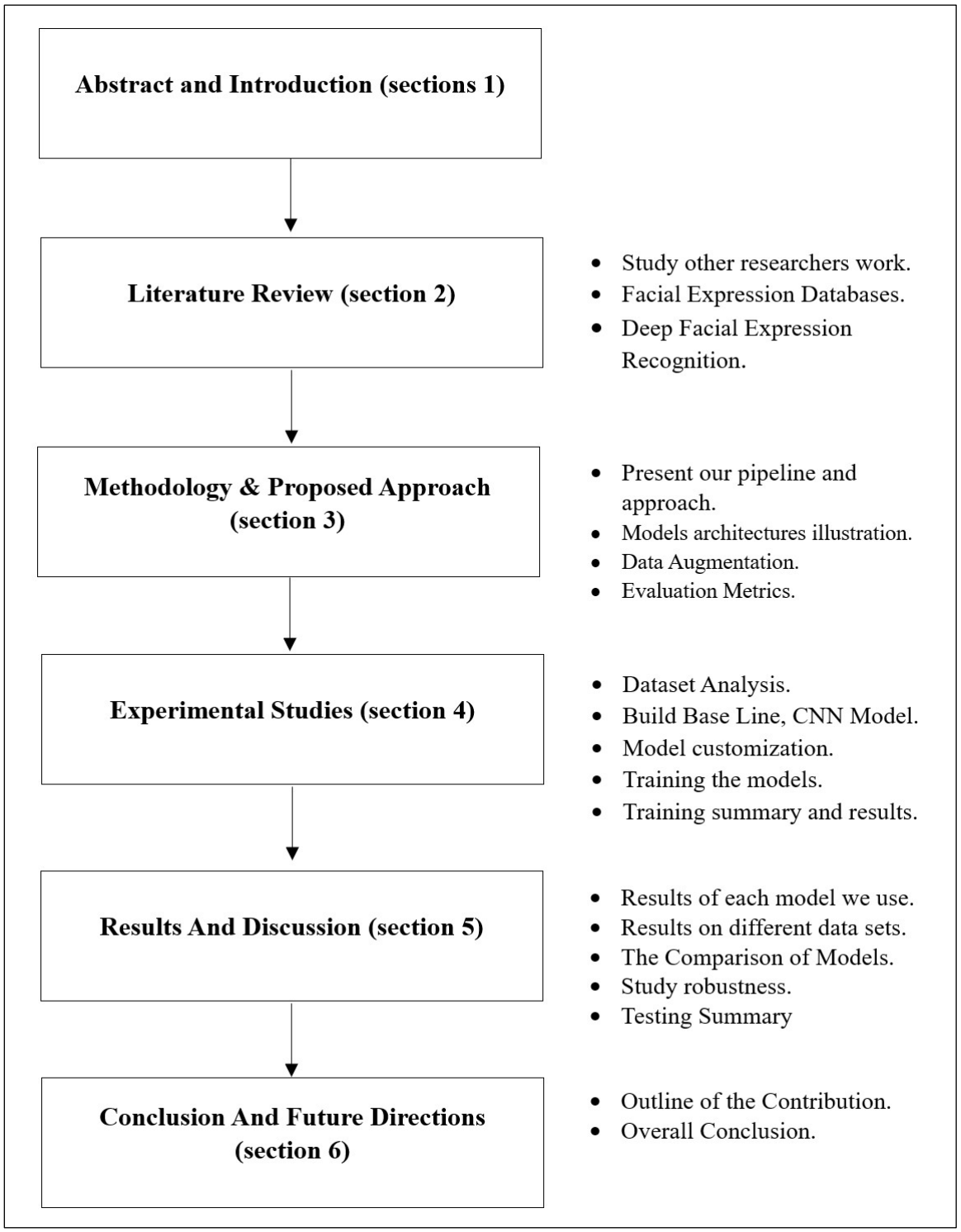


Figure 1.1. Thesis organization

2. LITERATURE REVIEW

Numerous representations of human emotion, encompassing fundamental facial expressions, action units, and valence-arousal, have been extensively utilized in emotion recognition studies over a prolonged period.

The potential of Convolutional Neural Networks (CNNs) in image processing became evident since their introduction in the late 1990s [21]. Comprising convolutional layers, pooling layers, and fully connected layers, CNNs emerged as effective tools for manipulating static images. However, training data and computational power limitations constrained their use until the 2010s. Subsequent advancements in computer power and the accumulation of larger datasets rendered CNNs more practical for feature extraction and image classification [22].

In recent years, deep neural network-based algorithms, particularly CNNs like Residual Neural Network (ResNet), VGGNet, and AlexNet [23], have proven successful in classifying images and extracting features. The Long Short-Term Memory (LSTM) network [24], an evolution of the Recurrent Neural Network (RNN), is employed for recording serial information in natural language processing and video analysis.

Recent studies have aimed to learn various facial behaviour tasks jointly using diverse emotion representations. Kollias et al. proposed FaceBehaviorNet [25], the first study considering the joint learning of all facial behaviour tasks within a comprehensive framework. Leveraging freely available emotion databases, they presented two methods for connecting training assignments.

The Aff-Wild2 dataset [26], introduced by Kollias et al., marked the first extensive in-the-wild database with annotations for the three primary behaviour tasks. Their work suggested multitasking learning models utilizing both aural and visual modalities, along with using ArcFace loss [eng2019arcface] for emotion identification.

Exploring non-overlapping annotations in multitasking learning datasets, Kollias et al. [27], [28] investigated task-relatedness and devised a unique distribution-matching strategy to facilitate information exchange between tasks by aligning the distributions of predictions. In

2020, the IEEE Conference on Face and Gesture Recognition and the First Affective Behaviour Analysis in the Wild (ABAW) Competition [29], co-located events, used the Aff-Wild2 dataset to advance state-of-the-art techniques for dimensional, categorical, and facial action unit analysis and recognition.

2.1. Facial Expression Databases

This section provides an overview of various open-source datasets utilized in emotion recognition, as shown in Table 2.1. Some datasets are based on the seven basic emotions (Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral), while others rely on valence and arousal values, continuous metrics ranging from -1 to 1, describing human emotions (as shown in Figure 2.1).

The FER2013 dataset [15], [16] comprises 48x48 pixel grayscale images of faces, automatically registered to ensure the face is centered and occupies a consistent amount of space in each image. The primary task involves categorizing each face based on the expressed emotion into one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). The training set includes 28,709 samples, while the public test set contains 7,178 samples.

Aff-Wild2 [26] annotates a total of 431 subjects across 539 videos, comprising 2,595,572 frames. Of these subjects, 265 are male, and 166 are female. The dataset is partitioned into train, validation, and test sections with 253, 71, and 233 subjects, respectively.

AffectNet [30] has gathered about 440,000 facial photos manually annotated from search engines. The training phase exclusively employs photos containing neutral expressions and the six fundamental emotions, amounting to approximately 280,000 images.

RAF-DB (The Real-world Affective Faces Database) [17] contains around 30,000 facial photos annotated with simple or complex expressions. For training, only 12,271 images marked with fundamental emotions were utilized.

RECOLA (Remote Collaborative and Affective) [31] database was introduced by Ringeval et al., featuring natural and impulsive emotions in the continuous domain (arousal and valence). The corpus includes four modalities: electro-dermal activity, electro-cardiogram,

auditory, and visual. It involves 46 subjects recorded in French, totalling approximately 9.5 hours of recordings. The dataset is divided into validation (15 subjects), training (16 individuals), and test (15 participants) sections, with balanced gender, age, and mother tongues among the classes.

AFEW (The Affect in the Wild) [32] dataset comprises dynamic temporal facial expressions extracted from real-world scenes in movies and reality TV shows. With a total of 1,809 videos, it consists of a training set (773 video clips), a validation set (383 video clips), and a test set (653 video clips). Notably, 114 out of 653 video clips in the test set are real TV clips, adding complexity to the challenge. The training and validation sets primarily consist of authentic movie records. The dataset includes 330 participants ranging in age from 1 to 77, annotated based on seven facial expressions (angry, disgusted, afraid, happy, neutral, sad, and surprised). Emotion tasks focus on categorizing the audio-visual content of each clip into the seven fundamental emotional groups.

AFEW-VA (The Affect in the Wild for Valence and Arousal) database recently introduced [33] extends the AFEW dataset by providing annotations for valence and arousal. This extension involves 600 video clips extracted from feature films, designed to replicate real-world scenarios with occlusions, diverse lighting conditions, and subject-free movements.

The video clips vary in length, ranging from brief sequences (10 frames or less) to longer ones exceeding 120 frames. The database includes per-frame annotations for valence and arousal, with more than 30,000 frames annotated for dimensional AFEW-VA prediction. The annotations are represented using discrete values within the [10, +10] range. This dataset serves the purpose of advancing research in dimensional affect prediction for both arousal and valence dimensions.

Table 2.1. Gesture recognition databases

Database	Labels	Num. of Frames	Num. of Videos
RECOLA[27]	valence-arousal (continuous)	345, 000	46
AFEW[28]	seven basic facial expressions	113,355	1809
AFEW-VA[29]	valence-arousal (discrete)	30,050	600
Aff-Wild2 [23]	valence-arousal (continuous)	2,595,572	539
FER2013[15]	eight basic facial expressions	28,709	Images
AffectNet[30]	seven basic facial expressions	280,000	Images
RAF-DB[17]	seven basic facial expressions	12,271	Images

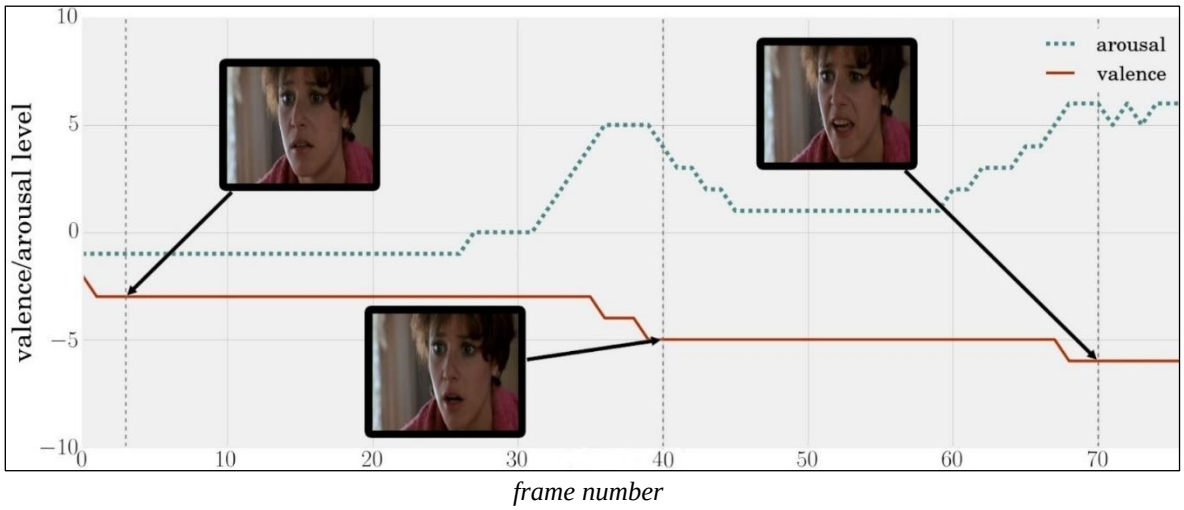


Figure 2.1. Annotations for valence and arousal over a section of a video, with the associated frames [33]

Emotional arousal signifies a state of heightened physiological activity, encompassing intense emotions like anger and fear. This state of emotional arousal is often triggered by daily experiences, such as the fight, flight, or freeze response.

Valence, on the other hand, refers to the pleasantness or unpleasantness of an emotional stimulus. Virtually all events and experiences, including faces, sounds, music, art, pictures, written or spoken language, can be classified along this dimension as more or less positive or negative.

2.2. Deep Facial Expression Recognition

Deep learning has recently demonstrated groundbreaking success across various application domains such as speech and image recognition [34,35]. In our context, we aim to leverage deep learning algorithms [36], among other techniques, for real-time facial expression recognition. This proposed system utilizes a camera to recognize and generate human emotions based on facial expressions, eliminating the need for manually constructed feature-based methods [37].

In this section, we explore different aspects of deep Convolutional Neural Networks (CNNs) for classifying the seven basic emotions or using a combination of CNN and Recurrent Neural Networks (CNN+RNN) to predict valence and arousal values. Some researchers exclusively utilize Deep CNN networks, while others introduce additional memory cells that integrate information from multiple frames to predict current arousal and valence. Each

approach is presented with its limitations and advantages.

In the FER2013 dataset, Table 2.2. shows the State-of-the-art results on FER2013 dataset. Zhang et al. [38] proposed a cross-dataset method for facial expression recognition. They incorporated three additional datasets AFLW, Celeb-Faces, and Kaggle alongside FER2013, leveraging facial attributes with associated labels. A bridging layer was created to connect the output with the FER2013 dataset, utilizing collective features from these datasets. The facial expression recognition accuracy achieved by their method was 70.6%. Devries et al. [39] developed a technique to assess the location and shape of facial landmarks, enhancing facial expression recognition capabilities. Their models include three fully connected convolutional layers, a fully connected ReLU hidden layer, and an output that employs the L2SVM activation function. Data augmentation techniques such as mirroring, rotating, zooming, and random photo rearrangement were applied. The method achieved a precision of 67.21%.

Table 2.2. State-of-the-art results on FER2013 dataset.

Models	Accuracy
Zhang et al. [38]	70.6%
Devries et al. [39]	67.21%

In the paper [40], the authors propose a novel Deep Locality-Preserving CNN (DLP-CNN) method designed to maximize inter-class scatters while preserving locality closeness to enhance the discriminative ability of deep features. Their approach yielded an impressive accuracy of 74.2% in the RAF-DB dataset.

In the RAF-DB dataset, Table 2.3. shows the State-of-the-art results on RAF-DB dataset. For example, Li et al. [41], the authors introduce a (CNN) with an attention mechanism (ACNN) designed to detect occluded facial parts and focus on the most discriminative unobstructed regions. ACNN employs a framework for holistic learning, combining representations from relevant facial regions (ROIs). A proposed gate unit calculates an adaptive weight for each region based on importance and unobstructedness, applying it to individual representations. Two ACNN variants are presented: patch-based ACNN (pACNN) considers only regional face patches, while global-local-based ACNN (gACNN) integrates global and local representations. The gACNN variant achieves the highest accuracy of 85.07% on the RAF-DB dataset.

In the paper [42], the authors propose a Region Attention Network (RAN) to adaptively capture the significance of facial areas for occlusion and position variant face emotion recognition. A backbone convolutional neural network generates various area features, which RAN aggregates and embeds into a short, fixed-length representation. They also suggest a region-biased loss to promote high attention weights for the most crucial regions, considering that facial expressions are primarily characterized by facial action units. The RAN model is evaluated on RAF-DB, achieving an accuracy of 86.90%.

Table 2.3. State-of-the-art results on RAF-DB dataset.

Models	Accuracy
DLP-CNN [40]	74.2%
gACNN [41]	85.07%
RAN [42]	86.90%

To assess the performance of the networks, two criteria were considered. The first criterion is the Concordance Correlation Coefficient (CCC), often used to evaluate the effectiveness of dimensional emotion identification techniques, as seen in the AVEC challenges. CCC evaluates the level of agreement between two-time series (e.g., video annotations and predictions) by scaling the correlation coefficient with the mean square difference. Predictions that are strongly correlated with annotations but have changed in value are penalized proportionally to the divergence. The CCC takes values in the range [-1, 1], where +1 implies perfect concordance and -1 denotes perfect discordance. Higher CCC values indicate a better fit between annotations and predictions, and high values are preferred.

The second criterion is the Mean Squared Error (MSE), the MSE provides a straightforward comparative statistic and offers us a general idea of how the derived emotion model is acting. A low MSE value is preferred. Below is a quick summary of the three papers that were accepted for the Aff-wild challenge. Tables 2.4 and 2.5 compares the results obtained (in terms of CCC and MSE) using all three approaches and the baseline network developed by the challenge organization. As one can see, the mean CCC and mean MSE for valence and arousal have been best achieved by FATAUVA-Net [43]. A deep convolutional residual neural network (ResNet) variation is initially described for the affective level estimation of facial expressions in the MM-Net approach [44]. Next, the temporal relationships between the video frames are modelled using various memory networks. Lastly, collective Multiple forecasts are combined using models. Memory networks demonstrate that the subsequent

stages enhance the initially attained performance, according to MSE greater than 10%.

A deep learning framework is described in the FATAUVA-Net technique [40], in which a core layer, an attribute layer, an action unit (AU) layer, and a valence-arousal layer are trained consecutively. Convolutional layers make up the core layer, followed by an attribute layer that extracts face traits. These layers are used to monitor how well AUs are learning. The intensity of valence and arousal is finally estimated using AUs as midlevel representations. Three neural network-based approaches that are based on Inception-ResNet modules and have been specifically developed for facial affect estimation are given and contrasted in the DRC-Net technique [45]. These techniques include Inception-ResNet with Long Short-Term Memory, Deep Inception-ResNet, and Shallow Inception-ResNet. Different scales of facial features are retrieved, and each frame simultaneously estimates both valence and arousal. Deep Inception-ResNet is the method that produces the best results for this paper.

The AffWildNet [46] consists of convolutional and pooling layers of either VGG-Face or ResNet-50 structures, followed by a fully connected layer and two RNN layers with GRU units. Valence and arousal are considered separately. To incorporate contextual information into the data, they developed a CNN-RNN architecture, in which the RNN part was fed with the outputs of either the first or the second fully connected layer of the respective CNN networks.

Table 2.4. Concordance Correlation Coefficient (CCC)

Models	Valence	Arousal	Mean
MM-Net [44]	0.196	0.214	0.205
FATAUVA-Net [43]	0.396	0.282	0.339
DRC-Net [45]	0.042	0.291	0.167
AffWildNet [46]	0.57	0.43	0.50

Table 2.5. Mean Squared Error (MSE)

Model	Valence	Arousal	Mean
MM-Net [44]	0.134	0.088	0.111
FATAUVA-Net [43]	0.123	0.095	0.109
DRC-Net [45]	0.161	0.094	0.128
AffWildNet [46]	0.08	0.06	0.07

3. METHODOLOGY & PROPOSED APPROACH

In this section, we present the framework of the proposed approach, as depicted in Figure 3.1. We used two open-source datasets: FER2013 [15-16] and RAF-DB [17]. The same pipeline is applied independently to each dataset.

Initially, the datasets are loaded and split into validation and training sets, with the test set pre-defined for each dataset. We employ a robust data loader capable of reading and shuffling the dataset when necessary, during training.

Different pipelines for augmentation techniques are employed to explore the training datasets and mitigate the risk of overfitting. This section discusses the applied augmentation pipeline and details each hyperparameter in use.

For building a robust model with reasonable accuracy, various deep-learning models are utilized: ResNet18[18], VGGNet16[19], VGGNet19[19], and EfficientNet-B0[20]. A baseline model is also developed from scratch, serving as a reference for subsequent development. This allows us to understand the impact of transfer learning and the use of common deep learning architectures. To delve deeper into exploration, the Genetic Algorithm is employed to design a CNN model architecture from scratch. The fitness function of the population models is chosen to be the Cross-Entropy loss of the designed models.

Following that, the models are trained and evaluated on the test set. A brief comparison between these models and approaches is presented based on the accuracy obtained on the test set. Additional metrics such as precision, recall, and F1 score for each model are calculated. Moreover, for a better understanding of the types of errors in the proposed models, the confusion matrix is computed for each trained model.

This section further elaborates on each model independently, illustrating the architecture of each model. Additionally, it details each evaluation metric used to compare the models based on these metrics and studies the performance of the proposed models on unseen data.

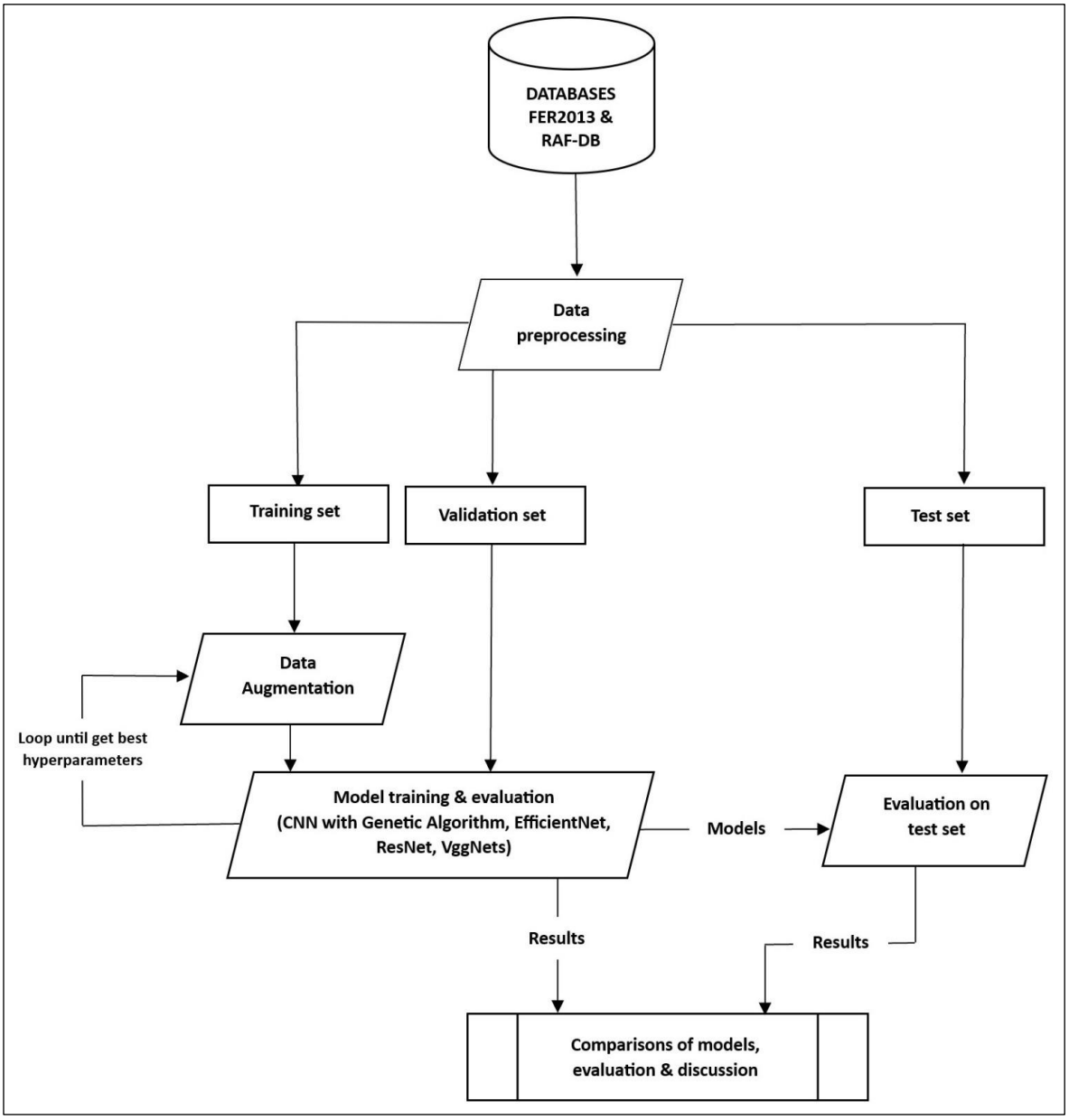


Figure 3.1. The pipeline framework

3.1. Convolutional Neural Network & Deep Learning Approaches

The Convolutional Neural Network (CNN) is a robust deep learning algorithm widely employed for image recognition, classification, and various applications. Unlike traditional models, CNN requires minimal pre-processing, as it autonomously learns filters without manual feature crafting. It processes images in small chunks, enhancing its ability to detect intricate patterns efficiently. The critical components of CNN include the input layer, output layer, and multiple hidden layers comprising Convolutional layers, Pooling layers (Max and Average pooling), Fully connected layers (FC), and normalized layers [47]. CNN utilizes

filters (kernels), weight arrays, to extract features from input images. Different activation functions are applied at each layer, introducing non-linearity and accommodating more complex data [48]. In the CNN process, the height and width decrease while the number of channels increases. The resultant column matrix is then utilized to predict the output [49], as shown in Figure 3.2.

Convolutional Neural Networks (CNNs) have demonstrated effectiveness across various visual tasks [50-51], and [52]. Each convolutional layer in CNN employs a set of filters that express spatial connection patterns within input channels. By interleaving convolutional layers with non-linear activation functions and down-sampling operators, CNNs can construct hierarchical picture representations, capturing spatial and channel-wise information. Pursuing more robust representations that emphasize essential image attributes for specific tasks, thereby enhancing performance, is a central theme in computer vision research. Recent advancements propose integrating learning mechanisms into CNNs to enhance representations by capturing spatial correlations between features. Noteworthy techniques, such as those popularized by the Inception family of designs [53], [54], involve incorporating multi-scale processes into network modules to boost performance. Further research has focused on refining models for geographical dependencies [22], [53] and integrating spatial attention into the network structure.

The effectiveness of VGGNets [19] and Inception models [55] has highlighted the significant impact of increasing network depth on the quality of learned representations. The introduction of Batch Normalization (BN) [56] has played a crucial role in stabilizing the learning process within deep networks, creating smoother optimization surfaces by managing input distributions across layers. Building on these insights, ResNets demonstrated that incorporating identity-based skip connections enabled the training of considerably deeper and more robust networks [18]. Introducing a gating mechanism to regulate information flow through shortcut connections and highway networks [7] further enhanced network capabilities. Subsequent studies have continued to refine the connections between network layers, resulting in promising advancements in the network's learning and representational properties.

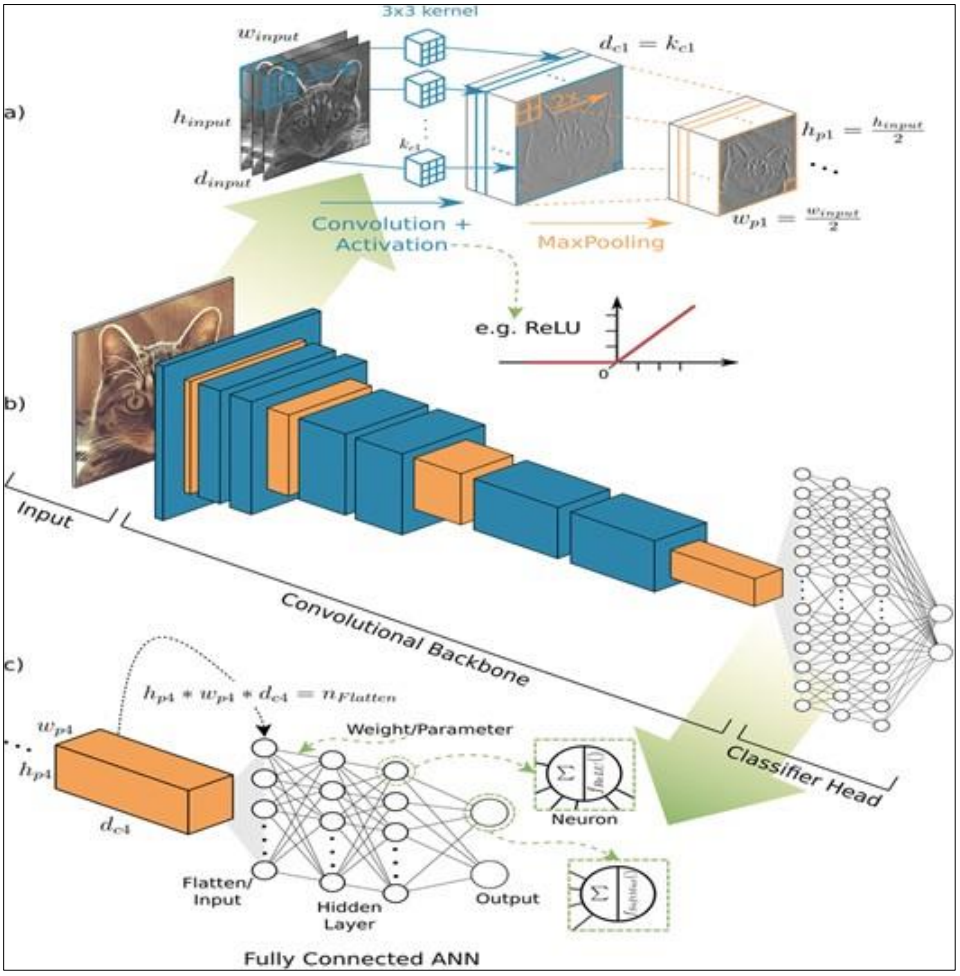


Figure 3.2. Overview and details of a convolutional neural network architecture for image recognition [57]

The convolutional backbone is the component of the network responsible for extracting features from the input image. The structure consists of multiple convolutional layers, where each layer applies a collection of filters to the input data. The purpose of these filters is to identify particular patterns in the image, such as edges, textures, and shapes. The result of every convolutional layer is a feature map, which is a 2D matrix of activations that indicate the intensity of the identified features at various positions in the image.

The convolutional backbone may also incorporate pooling layers, which decrease the dimensionality of the feature maps by subsampling them. This enhances the network's resilience to fluctuations in the input data and decreases the number of parameters that require learning. Classifier head of a fully connected artificial neural network:

The fully connected artificial neural network (ANN) classifier head is the component of the network that receives the output from the convolutional backbone and utilizes it to determine

a classification outcome. The structure consists of a sequence of interconnected layers, where each layer is a perceptron with a non-linear activation function.

The initial layer in the classifier head, which establishes connections between all neurons, is commonly referred to as the "flattened" layer. The purpose of this layer is to transform the output of the convolutional backbone, which is a 3D tensor consisting of height, width, and channels, into the 1D vector.

The subsequent fully connected layers in the classifier head receive this 1D vector as input and perform a sequence of linear transformations and non-linear activations. Typically, the last layer in the classifier head is a softmax layer that produces a probability distribution for the potential classes.

3.1.1. Genetic algorithm

The genetic algorithm (GA) is founded on evolutionary principles, drawing inspiration from natural selection and genetics. Initially developed by John Holland and popularized by David Goldberg [58], GA is an adaptive heuristic search method that utilizes bio-inspired operators such as mutation, crossover, and selection to generate optimal solutions [59].

Figure 3.3 depicts the flowchart of a genetic algorithm. The GA process begins with the random initialization of a population of individuals, representing CNNs with architectural variations. The fitness of each member is assessed based on its performance in specific image classification tasks, as measured by a deterministic fitness function. After evaluation, the selection procedure identifies individuals with the highest fitness. The crossover and mutation operators then generate new offspring from the selected parents. The fitness function evaluates the offspring, and the best members of the original population and its progeny form the new population, maintaining a constant size. The GA continues through a predetermined number of generations, with a maximum generation number serving as the termination criterion (e.g., 20 generations).

The utilized operators have a major impact on the genetic algorithm’s ability to converge (i.e., selection, crossover, and mutation). The operators are used to maintain genetic variability (mutation operator), produce new solutions by combining old ones (crossover

operator), and choose between solutions (selection). The following subsections provide explanations for each operator.

1) Selection: This operator determines which members of the population will produce new offspring in each generation. Various approaches for selection exist, with different strategies favouring different solutions.

2) Crossover: Creating a new individual by combining parts of two chosen individuals, the crossover operator is more likely to yield a better individual if the chosen parents are fit. Different strategies, such as single-point crossover, combine parts of the parent solution.

3) Mutation: The mutation operator prevents the genetic algorithm from converging to a local minimum and encourages genetic variation among individuals. It can substantially alter an individual, disrupting proximity among individuals and preventing premature convergence. Various mutation techniques, such as a simple permutation of the individual's encoding (e.g., swapping two layers in CNNs), can be applied.

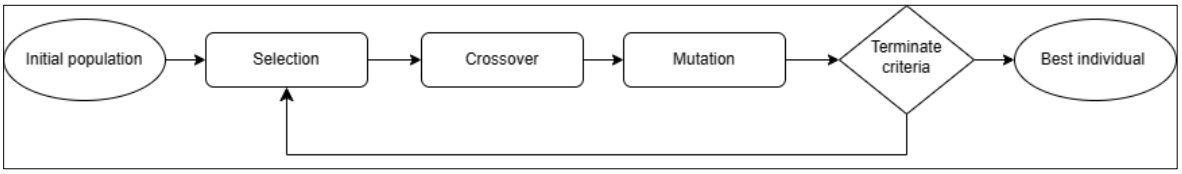


Figure 3.3. The flowchart of the genetic algorithm

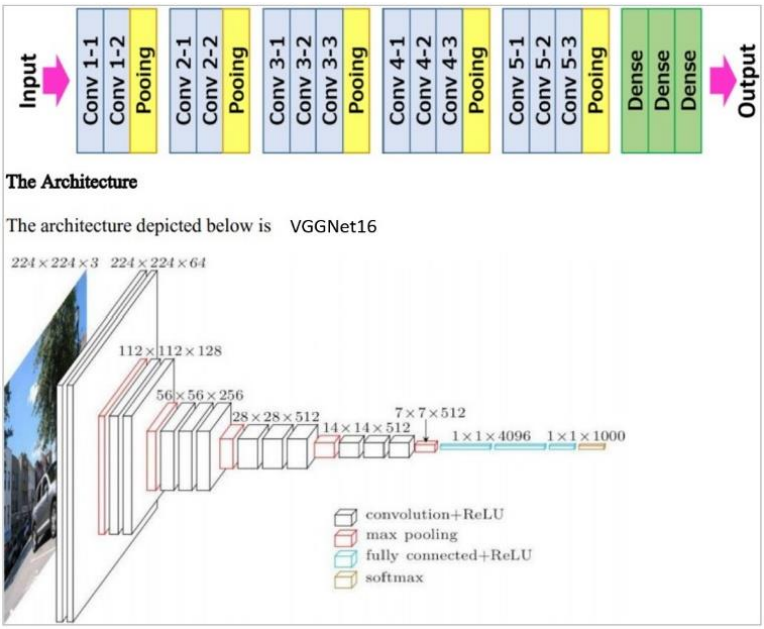
3.1.2. VGGNet

In the study "*Very Deep Convolutional Networks for Large-Scale Image Recognition*" [19], Simonyan and Zisserman from Oxford University introduced the convolutional neural network model known as VGG16. The model achieved a top-5 test accuracy of 92.7% in ImageNet, a dataset comprising more than 14 million images divided into 1000 classes. VGG16 became one of the well-known models submitted to ILSVRC-2014. Successively substituting several 3x3 kernel-sized filters for AlexNet’s larger kernel-sized filters (11 and 5, respectively, in the first and second convolutional layers) enhanced AlexNet. Using NVIDIA Titan Black GPUs, VGG16 underwent weeks of training.

Figure 3.3 (a) illustrates the architecture of VGGNet16. An RGB image with a defined size of 224x224 serves as the input for the Cov1 layer. The image is processed through

convolutional (Conv.) filters with a 3x3 receptive field, the smallest size capturing left/right, up/down, and center concepts. The 1x1 convolution filters, serving as the linear transformation of the input channels, are also used in one of the configurations. The spatial padding of the Conv. layer input is designed to retain spatial resolution after convolution, i.e., the padding is 1 pixel for 3x3 Conv. layers, and the convolution stride is fixed at 1 pixel. The five top layers, which are max-pooling layers, perform spatial pooling over a 2x2 pixel frame with a stride of 2. Following the stack of convolutional layers (with varying depths in different designs), three fully connected (FC) layers are present: the first two have 4096 channels each, while the third performs 1000-way ILSVRC classification and has 1000 neurons. The last layer is the softmax layer. In every network, the configurations of the fully linked layers are the same. Rectification (ReLU) non-linearity is the activation function for all hidden layers. Additionally, it should be noted that none of the networks, except for one, contain Local Response Normalization (LRN), which increases memory usage and computation time without improving performance on the ILSVRC dataset.

An overview of the VGGNet19 architecture, a convolutional neural network (CNN) used for image classification. VGGNet19 comprises 19 layers, including 16 convolutional layers and three fully connected layers. The convolutional layers are organized in groups of two or three, with max-pooling layers in between. VGGNet19 boasts a very deep architecture, enabling it to capture complex image features. The figure highlights the input and output sizes of each layer in the network, along with the number of filters and kernel sizes used in each convolutional layer, as shown in Figure 3.4 (b). Overall, VGGNet19 stands out as a robust CNN architecture widely employed in various computer vision tasks.



(a)



(b)

Figure 3.4. Overview of VGGNet's: (a) VGGNet16 (b) VGGNet19 architectures [60]

3.1.3. ResNet

ResNet, introduced by Kaiming et al. in 2016 [18], presents a novel approach to training deeper networks by utilizing residual learning. Conventional deep networks face challenges in training extremely deep models effectively. The innovation in ResNet lies in rethinking how layers in a network learn by focusing on residual functions. Based on empirical findings, it is observed that deeper neural networks, although advantageous for learning, encounter challenges in training and eventually reach a point of accuracy saturation, followed by a rapid decline. Residual learning addresses this challenge by enabling the training of deeper networks while avoiding the issue of accuracy degradation. In simple networks, the desired mapping is immediately learned by stacking several layers together. In contrast, the layers in residual networks are stacked to learn a residual mapping. Several models fit the mapping function, designated by the symbol H(x). According to the theory behind residual learning,

if numerous nonlinear layers can asymptotically estimate a challenging mapping function, they can do the same for the residual function, designated by the symbol F(x). The fundamental mapping is provided by:

$$H(x) = F(x) + x \tag{3.1}$$

Rather than learning the original function H(x), the stacked layers explicitly learn the residual function F(x). According to this approach, optimizing the residual mapping function is more manageable than optimizing the original function. The residual can be quickly driven to zero, mimicking an identity mapping. The original mapping function is then approximated as $H(x) = F(x) + x$. The residual shortcut link in a feed-forward neural network is implemented through this element-wise addition, approximating identity mapping. These connections in a residual network resemble identity mappings and do not introduce additional complexity or parameters to the networks. The training of these residual networks can be efficiently performed using SGD-based back-propagation.

Figure 3.5 illustrates the original architecture of ResNet-18. The network comprises a total of 18 layers, including 17 convolutional layers, 1 fully connected layer, and 1(softmax layer) for classification. The design ensures that when the output feature map has the same size as the convolutional layers' 3x3 filters, the layers also have the same number of filters. However, if the output feature map is halved, the number of filters is doubled in the layers. Two-stride convolutional layers are employed to downscale the data. The final two layers consist of average pooling, a fully linked layer, and a softmax layer. Shortcut connections are inserted between levels throughout the network. Two types of connections are used: the first type, represented by solid lines, employs identity mapping when the input and output sizes are the same, while the second type, depicted by dotted lines, utilizes connections for dimension expansion. The second type maintains identity mapping but with zero padding for larger dimensions and a stride of 2.

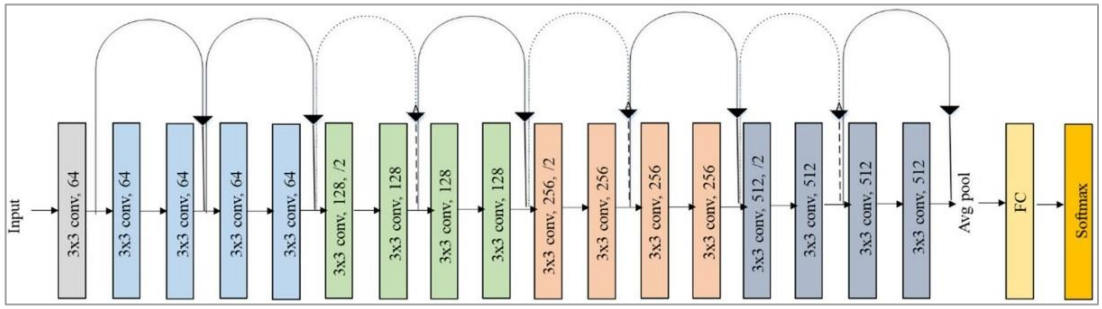


Figure 3.5. Overview of ResNet-18 architecture [61]

3.1.4. EfficientNet

Tan et al. [20] recently explored the relationship between the width and depth of CNN models and devised a practical methodology to construct CNN models with fewer parameters while achieving higher classification accuracy. They introduced seven models, denoted as EfficientNetB0 through EfficientNetB7, demonstrating superior performance in terms of the number of parameters and top-1 accuracy when evaluated with the ImageNet dataset [18].

EfficientNet's scalability revolves around a novel approach that uses a simple yet highly effective compound coefficient. Unlike traditional methods that independently scale network dimensions such as width, depth, and resolution, EfficientNet uniformly scales each dimension with a set of given scaling coefficients. While scaling individual dimensions can enhance model performance, EfficientNet optimally balances all network characteristics, efficiently leveraging available resources to enhance overall network performance.

At the core of the EfficientNet model family is the mobile inverted bottleneck convolution (MBConv), drawing inspiration from ideas in the MobileNet models [63]. This includes the use of depth-wise separable convolutions, combining pointwise and depth-wise convolution layers in sequence. Additionally, EfficientNet incorporates concepts from MobileNet-V2, such as inverted residual connections and linear bottlenecks.

EfficientNet integrates the depth-wise convolutions and linear activation function in bottleneck layers from MobileNetV2. The layer highlighted in red in Figure 3.7, labeled as a bottleneck layer, employs a linear activation function due to the channel bottleneck at various network parts. The authors argue that the commonly used ReLU activation function,

which discards values less than zero, does not perform well with inverted residual blocks. For the layer with fewer channels (the bottleneck channel), using a linear activation function yields better performance.

Furthermore, this network introduces a novel activation function called Swish [63], replacing the ReLU activation function. The Swish activation function offers some performance advantages similar to ReLU and LeakyReLU functions, as illustrated in Figure 3.6. It is a smoother activation function compared to these Formally, the Swish function is defined by the equation:

$$f_{Swish}(x) = \frac{x}{(1+e^{-\beta x})} \tag{3.2}$$

Where $\beta \geq 0$ is a parameter that can be learned during training of the CNN model. Note, if $\beta = 0$, f_{Swish} becomes the linear activation function and as ($\beta \rightarrow \infty$), f_{Swish} looks more and more like the ReLU function except it is smoother as shown in Figure 3.6. The Swish has a similar shape but is smoother [63]. The first observation is that the MBConv1, MBConv3, and MBConv6 blocks in this baseline model are repeated. These are essentially several MBConv block kinds. The second finding is that there are more channels or extended channels inside each block (due to more filters). The final finding is the presence of inverted residual connections between the model's thin layers. The squeeze-and-excitation (SE) idea was also incorporated by the authors in [65] into the MBConv blocks, which further boosts performance. Figure 3.8 illustrates the SE concept.

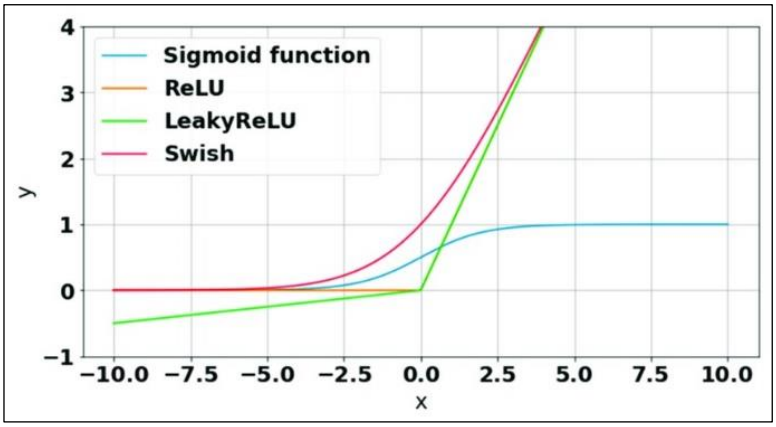


Figure 3.6. The new Swish [63] activation function, compared to ReLU and LeakyReLU

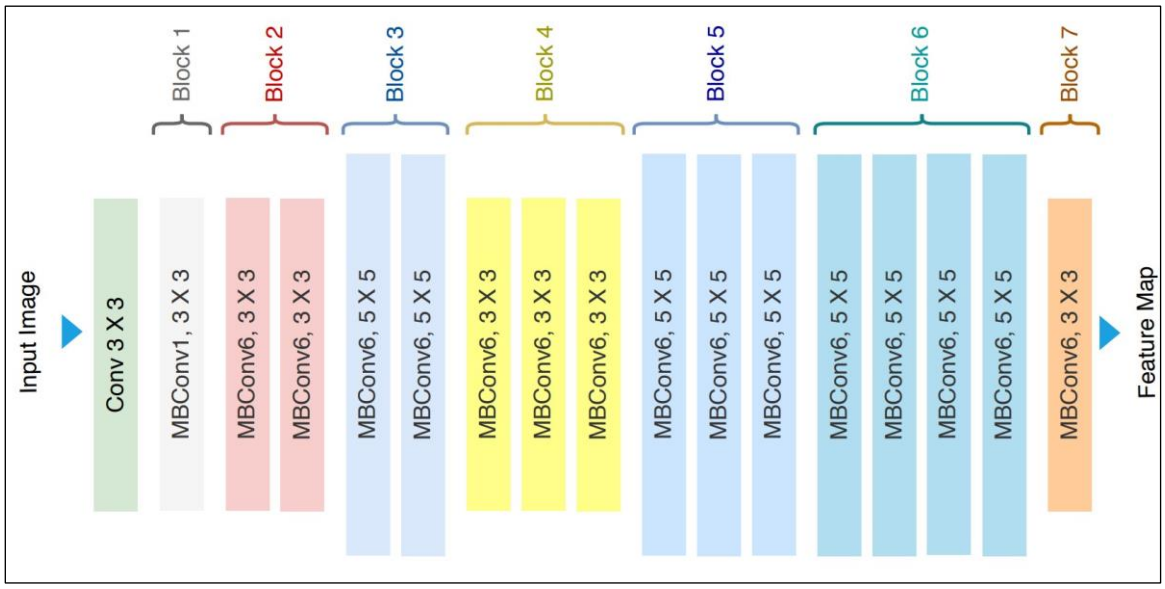


Figure 3.7. Overview of the EfficientNet-B0 model [64]

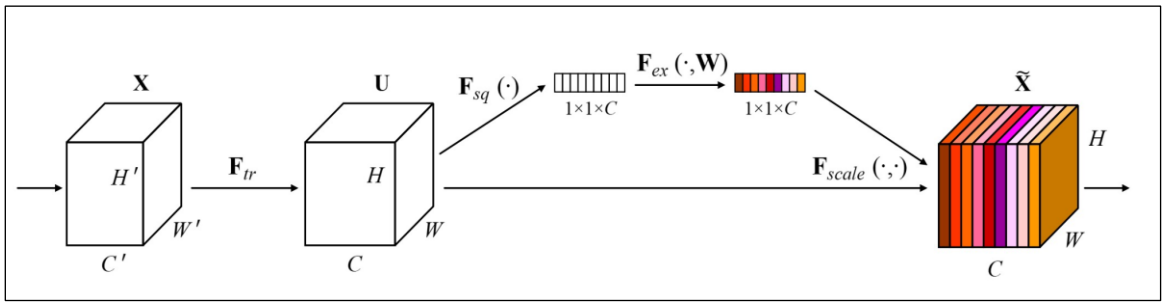


Figure 3.8. Illustration of the squeeze-and-excitation concept [65]

Squeeze-and-Excitation (SE) block, a type of convolutional neural network (CNN) architecture. The SE block is designed to improve the performance of CNNs by enabling the network to learn which features are important and which are not.

The symbols in the image are as follows:

X: This represents the input tensor to the SE block.

H, W, C: These symbols denote the height, width, and number of channels of the input tensor, respectively.

U: This symbol represents the squeeze operation, which reduces the spatial dimensions of the input tensor.

F_sq: This is the squeeze function, which is applied after the squeeze operation.

F_ex: This is the excitation function, which is applied after the squeeze function to learn channel-wise dependencies.

F_scale: This is the scaling function, which is applied after the excitation function to scale the features.

F_r: This is the reduction function, which is applied after the scaling function to reduce the dimensionality of the features.

F_c: This is the convolution function, which is applied after the reduction function to learn spatial dependencies.

Y: This represents the output tensor of the SE block.

The SE block operates by first applying the squeeze operation to the input tensor, then applying the squeeze function, the excitation function, the scaling function, the reduction function, and finally the convolution function, to produce the output tensor. This process allows the SE block to learn and emphasize important features while suppressing less important ones. This makes the SE block a powerful tool for enhancing the performance of CNNs.

3.2. Data Augmentation

We must artificially enlarge our dataset to get around the overfitting issue. We can increase the size of the dataset we already have using Data Augmentation. The goal is to replicate the differences seen when someone takes a picture or video by making modest changes to the training data.

Data augmentation methods modify the training data by altering the image pixel representation while maintaining the original label. There are a variety of methods that can be used for augmentation such as horizontal and vertical flips, color hiccups, random crops, translations, and rotations, and there are many other augmentations that are used frequently.

We can easily double and expand our training examples and build a more robust model by applying only a few of these training changes that can notably improve our results. Data augmentation is a technique used to increase the size of a training dataset by creating modified copies of the original data.

Here is an example of data augmentation with the following parameter ranges:

1) Random resize crop: It is a method of torch-vision. transforms module is used to crop a random area of the image and resize this image to the given size. This method accepts both PIL Image and Tensor Image. The tensor image is a PyTorch tensor with [C, H, W] shape, where C represents a number of channels and H, W represents height and width. This method returns a randomly cropped image. Here we used the scale (0.8, 1.2) is used which means the image will zoom by 80% and zoom out 120% while keeping the original size (48x48).

2) Adjust brightness: We randomly change the brightness, contrast, and saturation of an image. We used (brightness=0.5, contrast=0.5, saturation=0.5,) with 50% probability (p=0.5).

3) Random Affine: The method accepts PIL image and tensor image. The tensor image is a PyTorch tensor with [C, H, W] shape, where C represents the number of channels, and H and W represent the height and width, respectively. This method returns the affine transformed image of the input image. We used (0, translate=(0.2, 0.2)), with 50% probability (p=0.5).

4) Horizontal Flip: Also known as image mirroring, this is a technique that involves flipping an image horizontally, resulting in a mirrored version of the original image. This transformation essentially swaps the left and right sides of the image, creating a mirror image effect with a 50% probability.

5) Random Rotation ranges from [-45,45] degrees.

6) Tencrop: After all the aforementioned transforms are applied this method takes the image and Returns: a tuple of 10 cropped PIL Image or tensor. we used (40,40) crop dimensions.

7) Normalization: Image transformation is a technique that modifies the original pixel values of an image to a new set of values. A common type of image transformation is converting an image into a PyTorch tensor. This transformation scales the pixel values between 0.0 and 1.0 when converting the PIL image with a pixel range of [0, 255] to a

PyTorch FloatTensor of shape (C, H, W) with a range [0.0, 1.0]. We can use torchvision.transforms.ToTensor() to perform this transformation in PyTorch. Normalizing the images is a good practice when we use deep neural networks. Normalizing the images means changing the images to have a mean and standard deviation of 0.0 and 1.0, respectively, for each channel. To do this, we subtract the channel mean from each input channel and then divide the result by the channel standard deviation.

8) Random erasing: A torch.tensor image has its pixels erased in a rectangular region that is chosen randomly.

Parameters:

p – the probability that the random erasing operation will be performed. We used the 50% probability.
scale – refers to the range of the proportion of erased area against the input image.
ratio – represents the range of the aspect ratio of the erased area.
value – erasing value. The default is 0. If a single int, it is used to erase all pixels. If a tuple of length 3, it is used to erase R, G, and B channels respectively. If str of 'random', erase each pixel with random values.
in place – boolean to make this transform in place. Default set to false.

This is just an example but in training, we apply small ranges of these parameters to be compatible with real-life images shown in Figures 3.10 and 3.11.



Figure 3.10. Before and after augmentation, the FER2013 dataset.

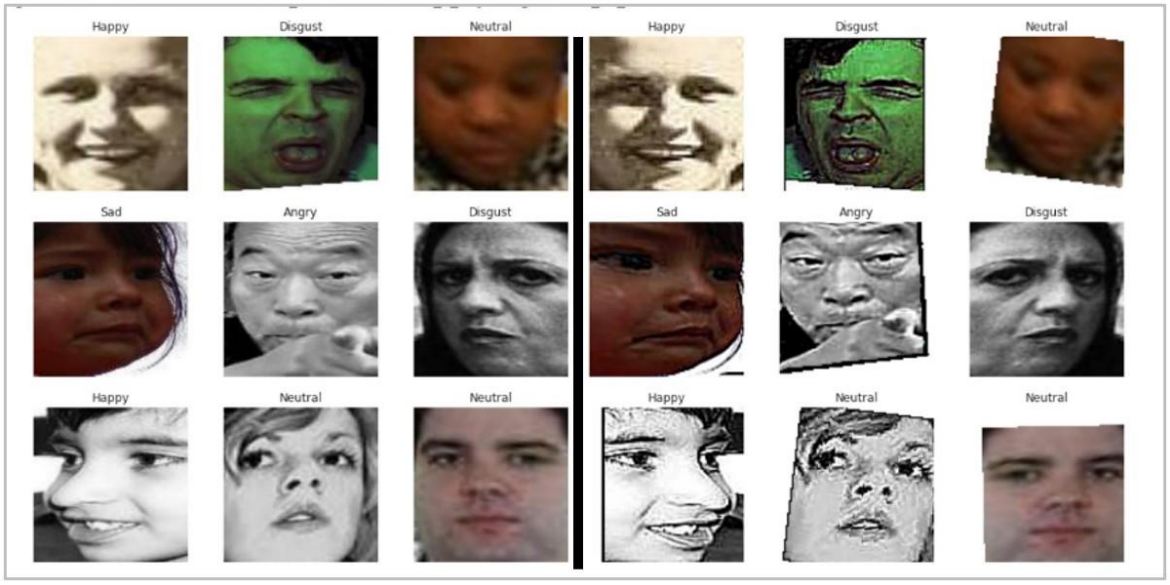


Figure 3.11. Before and after augmentation, the RAF-DB dataset.

3.3. Evaluation Metrics

Working with evaluation measures is mostly done to determine how well a machine learning model will perform on new data. For balanced datasets, metrics like accuracy, precision, and recall are useful ways to assess classification models. However, if the data is unbalanced, alternative techniques like ROC/AUC are more effective at assessing the model's performance.

The ROC curve, which is a complete curve that offers detailed information about the

classifier’s behavior, is more than just a single number. Additionally, it takes time to swiftly compare various ROC curves to one another.

Precision is helpful when the target class is well-balanced, but it may not be the most suitable option for imbalanced classes. Imagine a situation where our training data had 99% images of the dog but just 1% images of the cat. The dog would then always be predicted by our model, giving us 99% accuracy. Data is constantly unbalanced, as evidenced by spam emails, credit card fraud, and incorrect medical diagnoses. Therefore, other metrics like recall and precision should also be taken into account if we want to do a better model evaluation and have a complete picture of the model evaluation.

3.3.1. Confusion matrix

In Figure 3.12, you can observe that there are only two classes in a binary classification problem, ideally a positive and a negative class. Let’s now examine the Confusion Matrix’s metrics.

- The number of predictions in which the classifier accurately identifies the positive class as positive is known as True Positive (TP).
- The number of predictions in which the classifier accurately predicts the negative class as negative is known as True Negative (TN).
- False Positive (FP): This is the number of predictions in which the classifier pronounces the negative class as positive in error.
- The term "False Negative" (FN) describes the quantity of predictions in which the classifier misclassifies a positive class as negative.

It’s usually preferable to utilize the confusion matrix as your machine learning model’s evaluation criterion. It provides you with a very straightforward but effective performance measurement for your model.

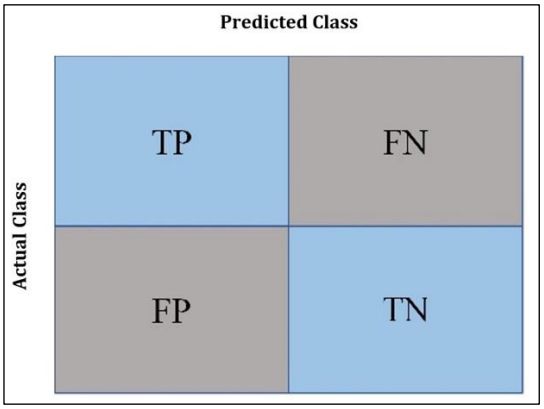


Figure 3.12. Confusion matrix for binary classification [66]

3.3.2. Receiver Operating Characteristic (ROC) Curve

The Receiver Operating Characteristic (ROC) curve is a graphical representation of the performance of a binary classification model. It is a plot of the true positive rate (sensitivity) against the false positive rate (1-specificity) at various threshold settings [66].

The area under the ROC curve (AUC) is a metric that provides a measure of the overall performance of a classification model. It represents the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance.

In simpler terms, the ROC curve and AUC help us evaluate how well a binary classification model is able to distinguish between positive and negative cases. A higher AUC value indicates better performance, with a perfect classifier having an AUC of 1.0. The ROC curve can also help us determine the optimal threshold for making classification decisions based on the specific requirements of a given application.

3.3.3. Accuracy

In machine learning, accuracy is a commonly used metric to evaluate the performance of a model. It measures the proportion of correct predictions made by the model. However, it's important to understand both its strengths and limitations to interpret it accurately.

Strengths: Simple and intuitive: Accuracy is straightforward to calculate and understand. It directly reflects the percentage of correct predictions, making it easy to interpret and communicate. Effective for balanced classes: When all classes have a similar number of examples, accuracy is a reliable indicator of the model's overall performance.

Useful for initial evaluation: It can be used as a quick and initial measure of a model's performance, particularly during the early stages of development.

Limitations: Bias towards majority class: In imbalanced datasets, where some classes have significantly more examples than others, accuracy can be misleading. A model predicting the majority class most of the time can achieve high accuracy even if it performs poorly on the minority class.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.3)$$

True Positives (TP) represent correctly identified positive instances, while True Negatives (TN) denote accurately identified negative instances. False Positives (FP) signify incorrectly identified positive instances, and False Negatives (FN) indicate incorrectly identified negative instances.

3.3.4. Recall

Recall describes how many of the actual positive cases our model was able to properly anticipate. When a False Negative is more important than a False Positive, it is a valuable metric. It is crucial in medical situations because even if we raise a false alarm, the real positive examples shouldn’t go undetected.

$$Recall = \frac{TP}{TP+FN} \quad (3.4)$$

Recall for a label is defined as the number of true positives divided by the total number of actual positives.

3.3.5. Precision

Precision reveals how many of the situations that were predicted with accuracy ended up being positive. When false positives are more problematic than false negatives, precision is helpful. Precision is essential for e-commerce websites, music or video recommendation systems, and other applications where inaccurate results could cause customers to leave, which would be bad for business. Precision for a label is defined as the number of true

positives divided by the number of predicted positives.

$$Precision = \frac{TP}{TP+FP} \tag{3.5}$$

3.3.6. F1-score

It provides a synthesis of the Precision and Recall measurements. It reaches its optimum when Precision and Recall are equal. The F1 score is the harmonic mean of precision and recall. More high values are penalized by the F1-score. F1 score may function as a useful evaluation statistic in the following circumstances:

$$F1\ score = \frac{2\ x\ precision\ x\ recall}{precision+recall} \tag{3.6}$$

- When FP and FN are equally costly.
- Adding more data doesn't effectively change the outcome.
- TN is high.

4. EXPERIMENTAL STUDIES

In this section, the training modules are discussed and go further with the training process of each employed model. Adam optimizer is used for hyperparameter optimization during training [67]. Adam optimizer involves a combination of two gradient descent methodologies: a. Momentum: This algorithm is used to accelerate the gradient descent algorithm by taking into consideration the exponentially weighted average of the gradients. Using averages makes the algorithm converge toward the minima at a faster pace. Also, a learning rate scheduler mechanism is used. Learning rate schedules seek to adjust the learning rate during training by reducing the learning rate according to a pre-defined schedule; b. Reference model: A baseline model is designed from scratch to be a reference for the next development of the Genetic Algorithm, and Deep Learning models ResNet18 [18], VGGNet16 [19], VGGNet19 [19], and EfficientNet [20]. We use transfer learning to train these three Deep CNN architectures pre-trained on the ImageNet dataset. Transfer learning means the use of previously acquired knowledge and skills in new learning or problem-solving situations. Thereby similarities and analogies between previous and actual learning content and processes may play a crucial role.
These models are customized to be capable of training on our seven basic emotions for both datasets FER2013 and RAF-DB. For each training section for each model, we will illustrate our custom layers and discuss their effect.

The following sections show the training and validation accuracy and cross-entropy loss of each employed model, and in the next section, all these models are evaluated on the test set. The result we obtained using ResNet18, VGGNet16, VGGNet19, and EfficientNet on the FER2013 dataset makes us only consider these approaches on the other dataset RAF-DB.

4.1. Dataset Analysis

The face is the most informative part of the body and its most important part for gesture recognition. Starting with a dataset from Kaggle containing grayscale images of size 48*48, this image has so much noise and is a challenging dataset because it's hard to get high accuracy on.

The FER2013 [15], [16] dataset consists of grayscale images of faces measuring 48 × 48 pixels. The faces have been automatically registered such that each one is roughly in the same location and takes up a similar amount of space. Each face must be assigned to one of seven categories, with 0 denoting anger, 1 disgust, 2 fear, 3 happiness, 4 sadness, 5 surprises, and 6 neutralities. The public test set has 7178 cases, whereas the training set has 28,709 examples.

RAF-DB [17] consists of around 30,000 facial photos with simple or complex expression annotations that can be found in the Real-world Affective Faces Database (RAFDB). Only 12,271 marked with fundamental emotions were used in the training portion.

The dataset contains seven labels distributed like in Figures 4.1 and 4.2. The Disgust label contains fewer images than other labels in the FER2013 dataset, which may be challenging in the learning part. In RAF-DB, fear labels include fewer images than other labels; the two datasets have a rich image of the happy label as in the two figures.

Figures 4.3 and 4.4 show examples of the two datasets FER2013 with grayscale images and RAF-DB with RGB images; we show these samples to get more intuition about how our data looks.

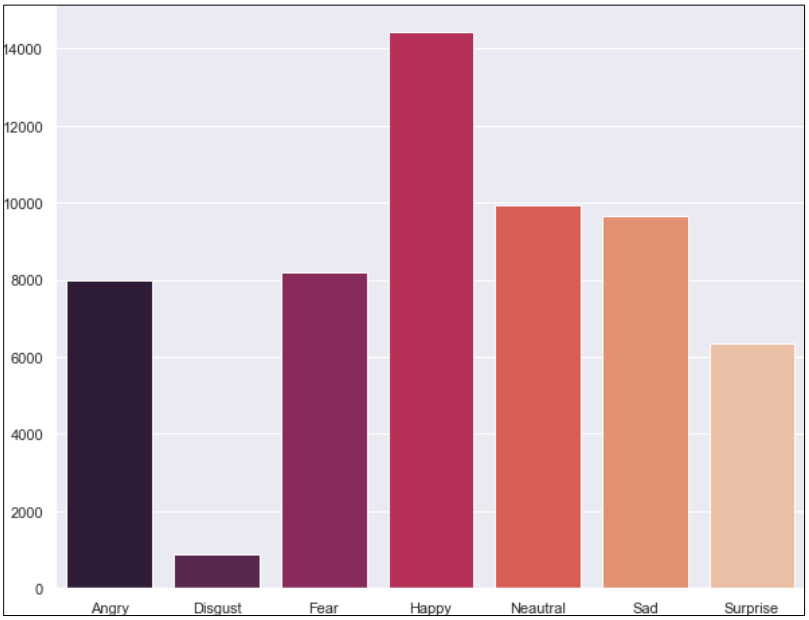


Figure 4.1. The illustration of the Kaggle's fer2013 dataset (label vs total number of training samples)

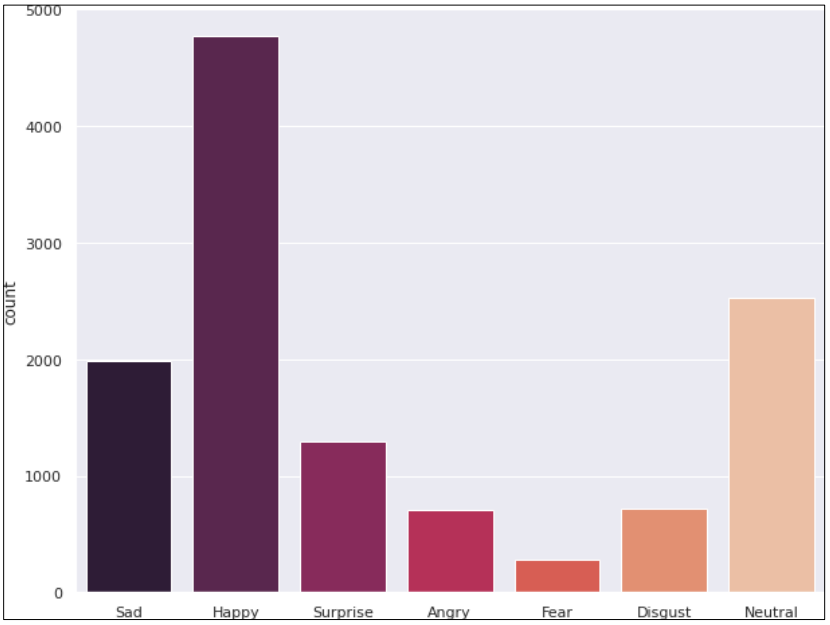


Figure 4.2. The illustration of the RAF-DB dataset (label vs total number of training samples)



Figure 4.3. Some grayscale samples from the FER2013 dataset

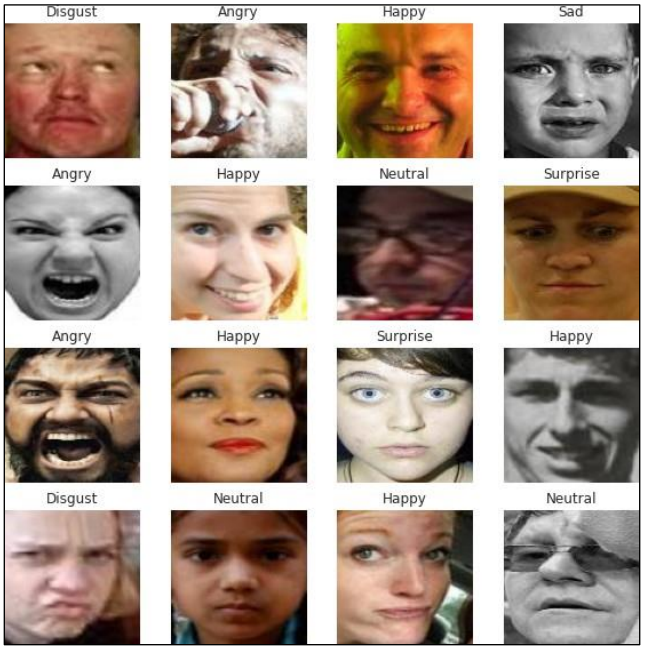


Figure 4.4. Some RGB image samples from the RAF-DB dataset

4.2. Build Baseline CNN Model

A baseline model is developed from scratch to be a reference for the next development of the Genetic Algorithm, and Deep Learning models ResNet18, VGGNet16, and EfficientNet in this section. The transfer learning used to train these three Deep CNN architectures pre-trained on the ImageNet dataset. Deep neural networks are more difficult to train, so a

residual learning framework is used to ease the training of networks. With the network depth increasing, accuracy gets saturated and then degrades rapidly. So, adding more layers doesn't mean improving the accuracy at all but the residual framework makes it easier for deeper layers to be trained easily. This residual block is used as a base building block for our architecture. This is the Keras plotting of the model architecture description of the connections and how many layers.

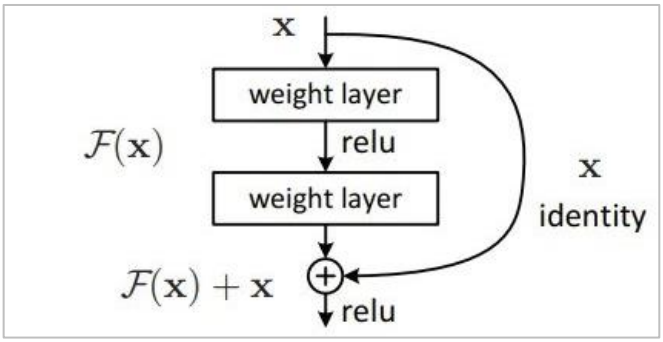


Figure 4.5. Residual learning: a building block

Figure 4.5 provides a general perspective on residual learning. The relevant components here are briefly explained below.

X: This denotes the input to the residual connection block. Usually, it is a tensor consisting of feature maps, with each map representing distinct characteristics of the input data, such as edges or textures.

F(x): This denotes the result of a convolutional layer within the residual block. The symbol "F" represents a universal operation applied to the input X, indicating the specific alteration carried out by the layer (such as feature extraction or filtering).

+: This symbol denotes the operation of adding corresponding elements together.

Identity (x): This denotes a replica of the input X. It functions as a detour route that enables the unaltered data to pass through the network in parallel with the modified attributes derived from F(x).

ReLU (Rectified Linear Unit): The rectified linear activation (ReLU) function is applied to the sum of F(x) and the ReLU activation function introduces non-linearity into the neural

network, enabling it to acquire a deeper understanding of intricate relationships within the data.

General structure**:** The residual connection block merges the modified features from F(x) with the initial input X using element-wise addition.

Figure 4.6 provides general architecture information for the baseline model based on the customized CNN. The features and benefits of this approach are mentioned below.

This design offers numerous benefits: Addresses the issue of vanishing gradients: By directly incorporating the input, the gradients can propagate more efficiently to preceding layers, thereby mitigating the problem of vanishing gradients that can impede the training of deep neural networks.

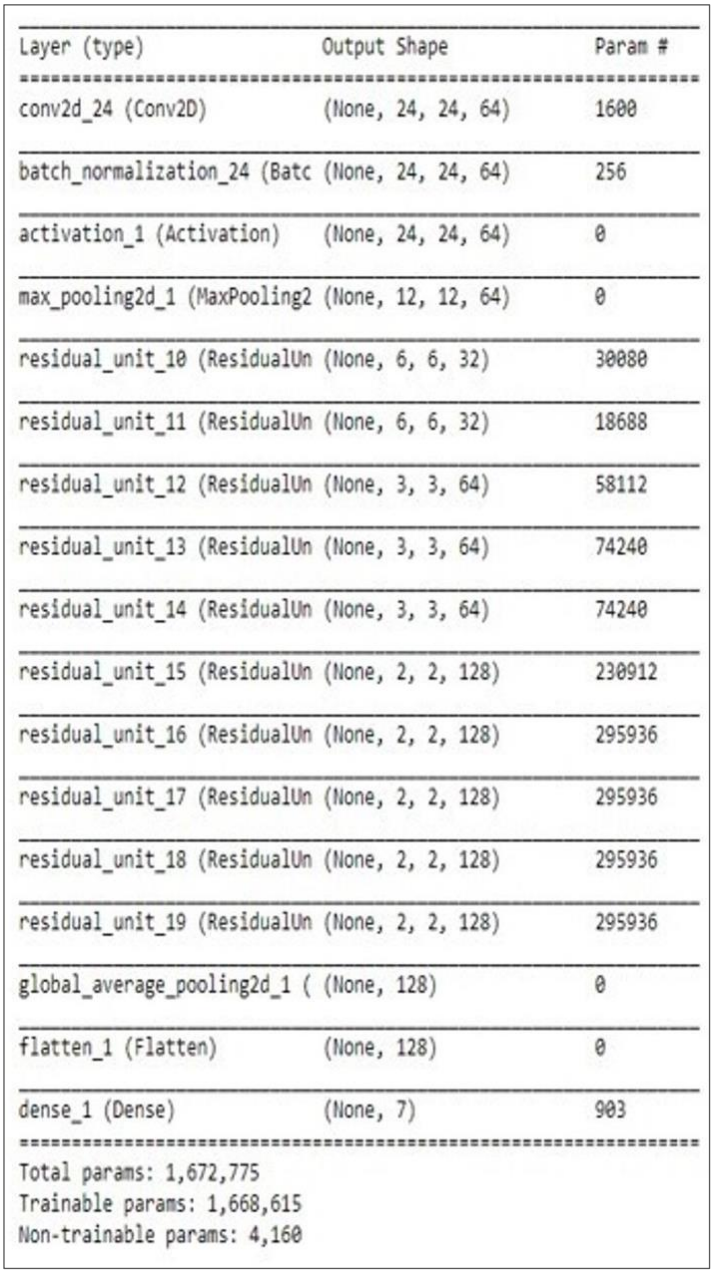
Enhances information transmission: Identity mapping enables the preservation of essential details in the original information, bypassing any potentially detrimental transformations in the F(x) path. This ensures that the network can effectively learn from the preserved information.

Improves model performance: Residual connections have been demonstrated to enhance the accuracy and generalization abilities of convolutional neural networks (CNNs) across different tasks.

Supplementary remarks: The precise characteristics of the F(x) function may differ based on the network architecture and the task. The process may incorporate several convolutional layers, pooling operations, and other non-linear functions.

Additionally, the performance can be further enhanced by stacking residual connection blocks within the network.

Summarization: The image provided illustrates a residual connection block implementing identity mapping, which is a widely employed technique in convolutional neural networks to enhance information propagation, mitigate the problem of vanishing gradients, and improve the overall performance of the model



Layer (type)	Output Shape	Param #
conv2d_24 (Conv2D)	(None, 24, 24, 64)	1600
batch_normalization_24 (Batc	(None, 24, 24, 64)	256
activation_1 (Activation)	(None, 24, 24, 64)	0
max_pooling2d_1 (MaxPooling2	(None, 12, 12, 64)	0
residual_unit_10 (ResidualUn	(None, 6, 6, 32)	30080
residual_unit_11 (ResidualUn	(None, 6, 6, 32)	18688
residual_unit_12 (ResidualUn	(None, 3, 3, 64)	58112
residual_unit_13 (ResidualUn	(None, 3, 3, 64)	74240
residual_unit_14 (ResidualUn	(None, 3, 3, 64)	74240
residual_unit_15 (ResidualUn	(None, 2, 2, 128)	230912
residual_unit_16 (ResidualUn	(None, 2, 2, 128)	295936
residual_unit_17 (ResidualUn	(None, 2, 2, 128)	295936
residual_unit_18 (ResidualUn	(None, 2, 2, 128)	295936
residual_unit_19 (ResidualUn	(None, 2, 2, 128)	295936
global_average_pooling2d_1 (	(None, 128)	0
flatten_1 (Flatten)	(None, 128)	0
dense_1 (Dense)	(None, 7)	903

Total params: 1,672,775
Trainable params: 1,668,615
Non-trainable params: 4,160

Figure 4.6. The baseline model information

4.2.1. Training the Baseline Model

The baseline model is trained for 300 epochs with an initial learning rate of 0.001 this hyperparameter tunning is used after an elusive search between different parameters. The model gets a test accuracy of 60.05% on the validation set. This is a good accuracy with respect to the data we are dealing with, which have grayscale images of 48x48 pixels. Figure 4.7 shows the accuracy development over the 300 epochs; also, Figure 4.8 shows the loss value of both the training and validation sets. The ROC AUC metric is also measured in Figure 4.9. The Reduce on Plateau Learning rate scheduler is illustrated in Figure 4.10. This scheduler reads a metrics quantity, and if no improvement is seen for a 'patience' number of epochs, the learning rate is reduced. Each epoch takes 2 minutes to train on the local CPU.

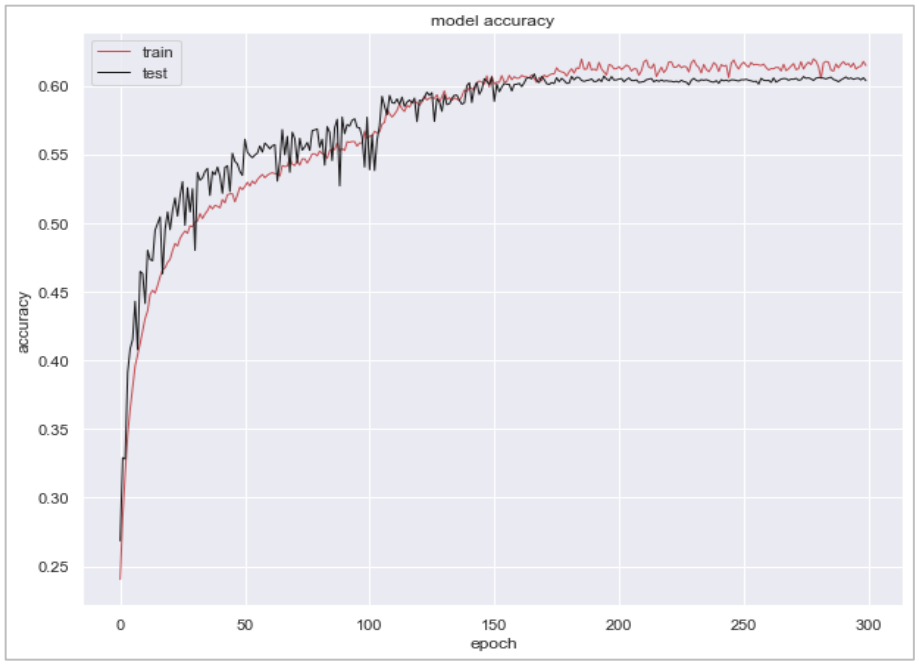


Figure 4.7. Training accuracy on the training set annotated with blue and the validation set annotated with orange

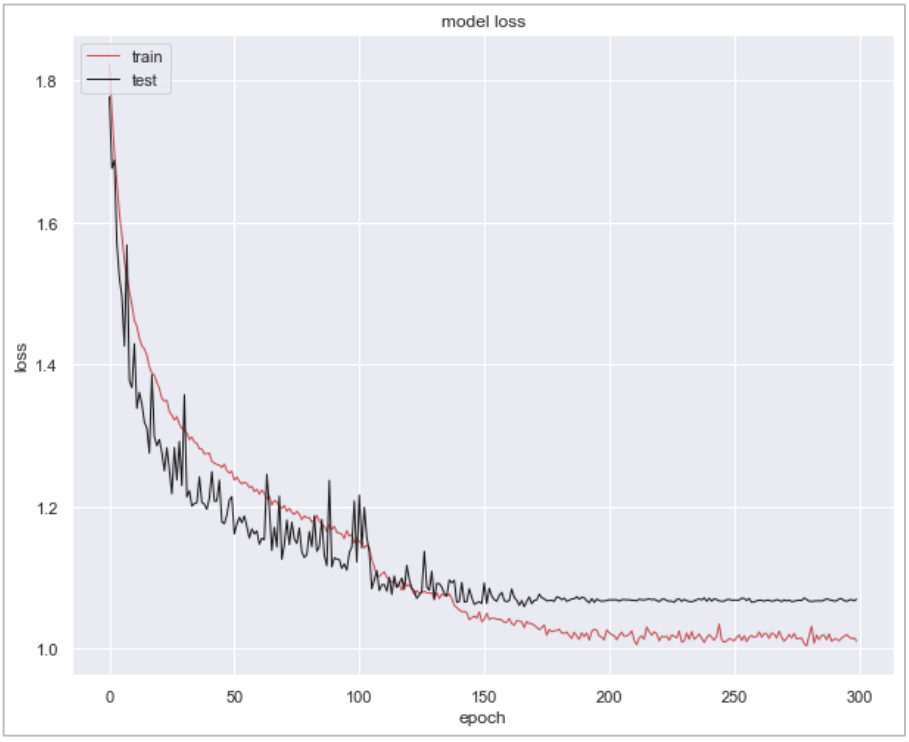


Figure 4.8. The loss of both the train and validation set, it's obvious that after 220, the loss stopped reducing, and the model can't reduce it more after reducing the learning rate many times. This also illustrates why the accuracy stopped improving after 220 epochs

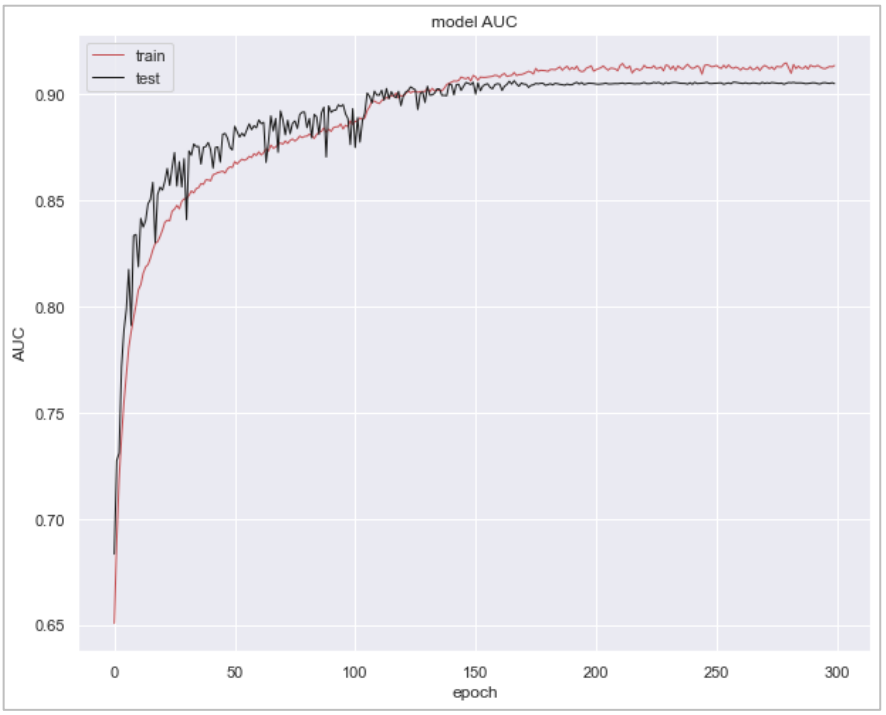


Figure 4.9. The AUC metric of the training and validation sets improved during the 300 training epochs

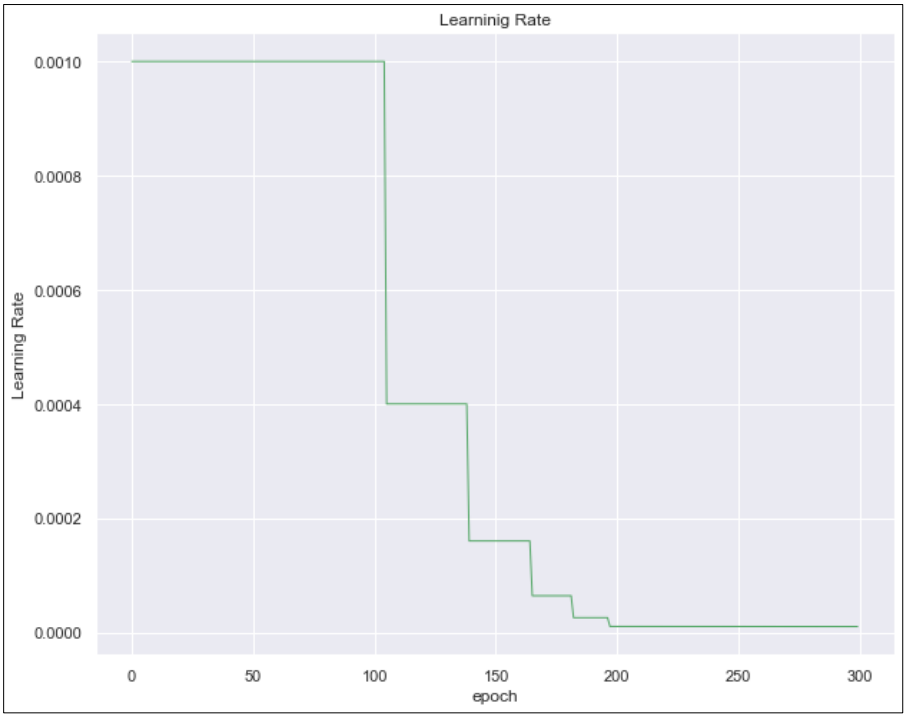


Figure 4.10. Reduce the Plateau Learning rate scheduler

4.3. Evolving Architecture for Convolutional Neural Networks Using Genetic Algorithm

In this section, the main steps of the proposed algorithm are outlined after presenting its framework. The suggested algorithm's flowchart is shown in Figure 6 (notice the variations from Figure 4.11).

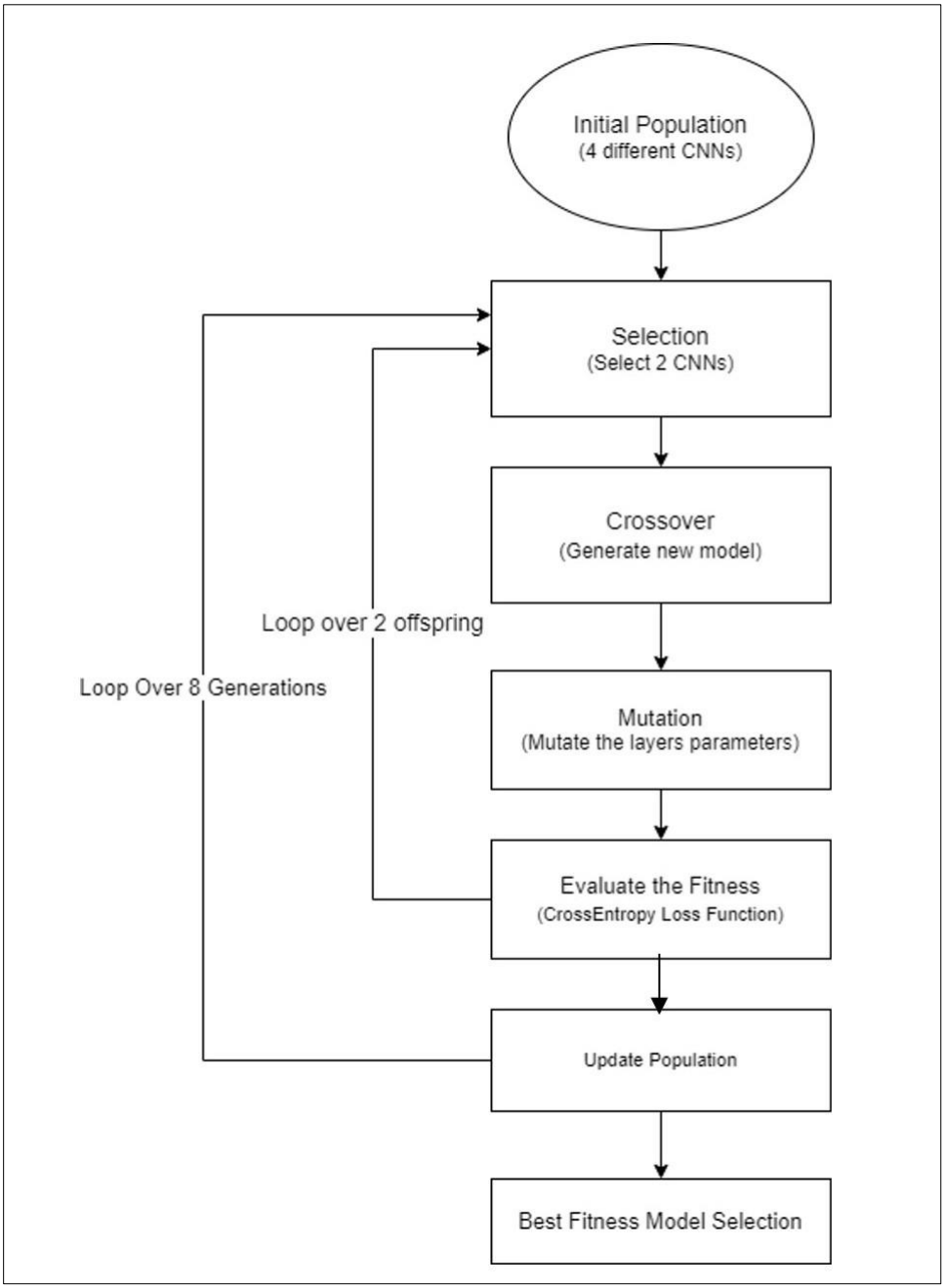


Figure 4.11. The flowchart of the proposed genetic algorithm

4.3.1. The overview of the used GA

The used method's structure [59] is described in Figure 4.12. The suggested algorithm starts to function specifically by providing the input CNN architecture, the population and offspring sizes, and the number of generations. Through a series of evolutionary processes, the algorithm eventually discovers an improved design for the input CNN.

First, the input CNN architecture is significantly mutated (lines 1–5) to compute the starting population. The evolution then works for the number of input generations (lines 6 through 15).

A new offspring is computed, assessed using the provided dataset, and compared to the population produced at the previous step at each stage of the evolution. In more detail, the generation of the new child involves choosing two individuals from the population (line 8), combining them using genetic operators (lines 9–10), and producing a person that encodes a specific architecture of the CNN. The new individual is next assessed (line 11) and contrasted with the population (line 12). The present population is used to calculate the number of individuals who will live in the next generation (line 14).

The suggested technique shows how a random search can be used intelligently. Although the proposed algorithm is not random, it works according to the problem-specific heuristic mechanism. It makes use of prior data to focus on the search process on areas of the search space that perform better than others.

The algorithm selects the layers of the CNN architecture across several generations. It gains knowledge through haphazard exploration and gradually starts to use what it knows to choose better models. It uses testing accuracy to compare various architectures and then chooses the best architecture. The procedure continues for numerous generations until a fully trained, appropriate CNN model is produced.

The success of the suggested algorithm depends on the parameter selection. The classification error of the output CNN is noticeably lower than that of the input CNN if the parameters have been properly set.

Algorithm 1 Framework of The Proposed Algorithm

Input: initial CNN architecture, population size P, , number of generations G, offspring size O.
Output: optimal CNN architecture.

```
for S ← 0 to P do
    i_s ← mutate the input CNN architecture;
    Calculate the fitness of i_s;
    P0 ← P0 ∪ {i_s}
end for
for t ← 0 to G do
    for k ← 0 to O do
        Select two individuals from P_t;
        Generate a new individual i_k using crossover;
        Randomly apply soft mutation to i_k;
        Evaluate the fitness of i_k;
        Evolve P_t with i_k;
    end for
    P_{t+1} ← P_t
        end for
        Select the best individual from P_t and decode it to
the corresponding CNN architecture.
```

Figure 4.12. The proposed (GA) approach [59]

4.3.2. Training genetic algorithm-designed models

By applying the genetic algorithm to find the optimal model that can fit the FER2013 dataset. We run the algorithm for 8 generations (G), 2 offspring (O), and with population size (P = 4). Table 4.1 shows the results of the optimal model of the population is Net1 with (372,423) parameters and minimum cross-entropy loss (1.46075). Figure 4.13 shows the optimal model architecture as follows:

- Layers and Types: The diagram shows several layers, including convolutional layers (Conv2D), max-pooling layers (MaxPooling2D), dropout layers, and a flatten layer. Each layer has a specific purpose in processing input data.
- Output Shape: For each layer, the output shape is indicated. For example, the first Conv2D layer has an output shape of (None, 48, 48, 32), which means it produces a four-dimensional tensor.
- Parameters: The number of parameters for each layer is listed. For instance, “conv2d_6” has 320 parameters. At the bottom, the total number of trainable parameters is given as 372,423.

After getting the optimal model, we continue its training for 300 epochs with a schedular learning rate using the Adam optimizer. Because the genetic algorithm only trains the model for 5 epochs and compares each individual's loss as its fitness function for each model. So, we want to train the model for more epochs to reach the best accuracy the model can have. Figure 4.14 shows the training accuracy curve, the model obtained 62.598% accuracy, with a validation loss of 0.9985. Also, Figure 4.15 shows the loss training curve. The graph represents the model's loss during the training process. It shows training and testing loss values over successive epochs. This plot visualizes how well the model is learning from the training data (train) and how well it generalizes its learning to unseen data (test) over successive epochs.

Table 4.1. Genetic algorithm final population results

Model	Loss (CE)	Num. of Parameters
Net_1	1.46075	372423
Net_2	1.46531	697063
Net_3	1.50992	719559
Net_4	1.53866	302663

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 48, 48, 32)	320
conv2d_7 (Conv2D)	(None, 46, 46, 64)	18496
max_pooling2d_3 (MaxPooling 2D)	(None, 23, 23, 64)	0
conv2d_8 (Conv2D)	(None, 23, 23, 64)	36928
max_pooling2d_4 (MaxPooling 2D)	(None, 12, 12, 64)	0
dropout_2 (Dropout)	(None, 12, 12, 64)	0
conv2d_9 (Conv2D)	(None, 12, 12, 64)	36928
conv2d_10 (Conv2D)	(None, 10, 10, 128)	73856
conv2d_11 (Conv2D)	(None, 8, 8, 64)	73792
max_pooling2d_5 (MaxPooling 2D)	(None, 4, 4, 64)	0
dropout_3 (Dropout)	(None, 4, 4, 64)	0
flatten_1 (Flatten)	(None, 1024)	0
dense_2 (Dense)	(None, 128)	131200
dense_3 (Dense)	(None, 7)	903

Total params: 372,423
Trainable params: 372,423
Non-trainable params: 0

Figure 4.13. Optimal GA architecture model

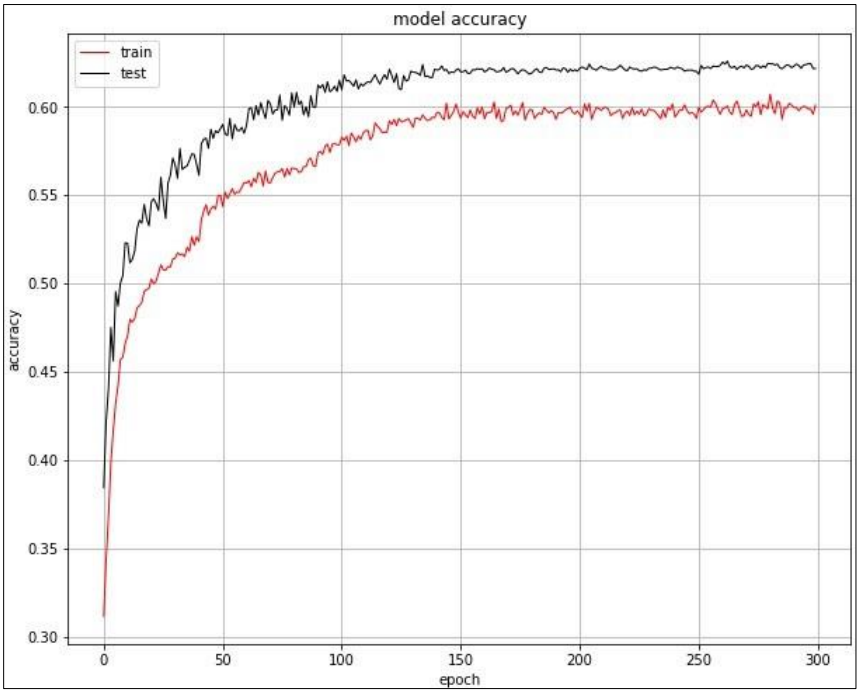


Figure 4.14. Training accuracy over training the model for 300 epochs with reduce on plateau learning rate scheduler

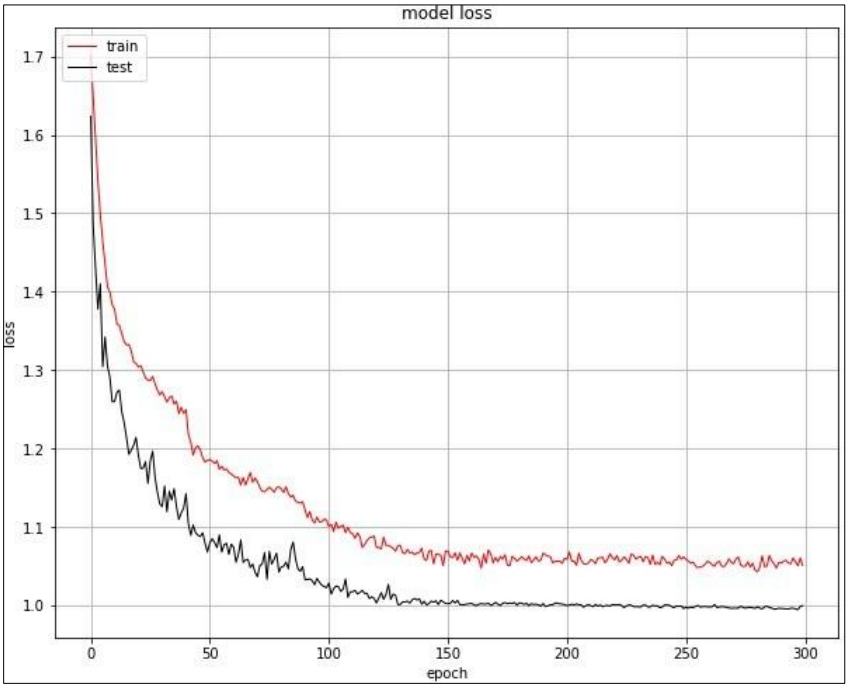


Figure 4.15. Training cross entropy loss curve

4.4. Training EfficientNet Model

In this chapter, the EfficientNet model is trained on two datasets FER2013 and RAF-DB. By customizing the model by changing the top layer, which consists of 1000 thousand neurons representing 1000 classes of the ImageNet dataset we replace this layer with just 7 neurons representing the seven basic emotions. We set the starting learning rate for 0.005 and use the scheduler learning rate to reduce the learning rate by a factor of 0.3 if the validation accuracy doesn't improve for the executive 5 epochs.

Figures 4.16 and 4.17 show the model accuracy and loss during the training on the FER2013 dataset. We stopped the learning after 40 epochs because the model started to overfit the training data, and the validation accuracy started to decrease. The model achieves the best validation accuracy of 67.09%; the training begins by 49% on the validation set after the first epoch because of transfer learning. Figures 4.18 and 4.19 show the model accuracy and loss during the training on RAF-DB. We also stopped the training after 40 epochs because the model started to overfit the training set too much. In this dataset, the model achieves the validation accuracy of 84.547%.

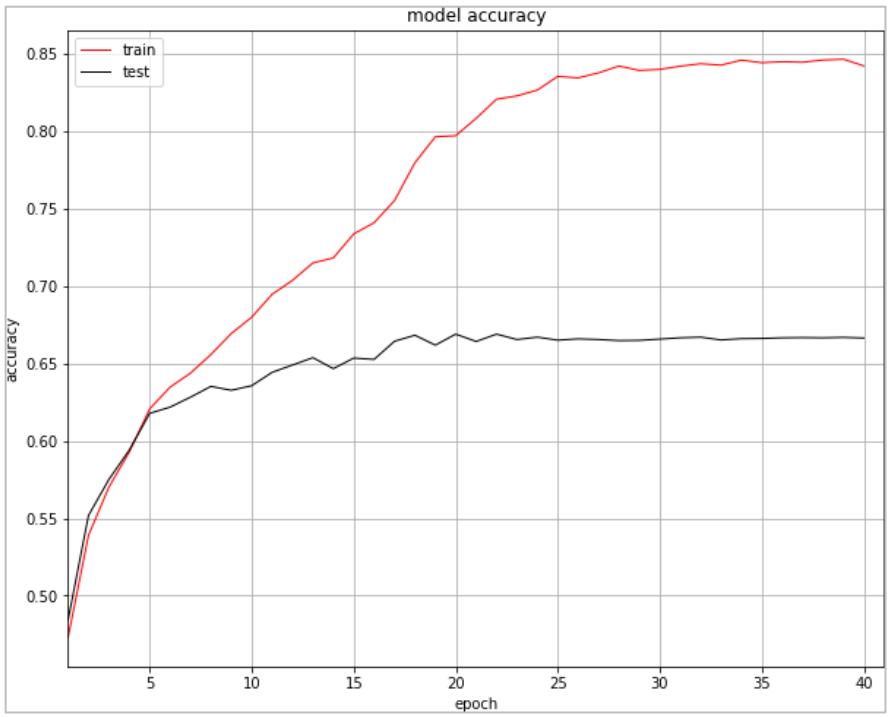


Figure 4.16. Training accuracy versus 40 training epochs on the FER2013

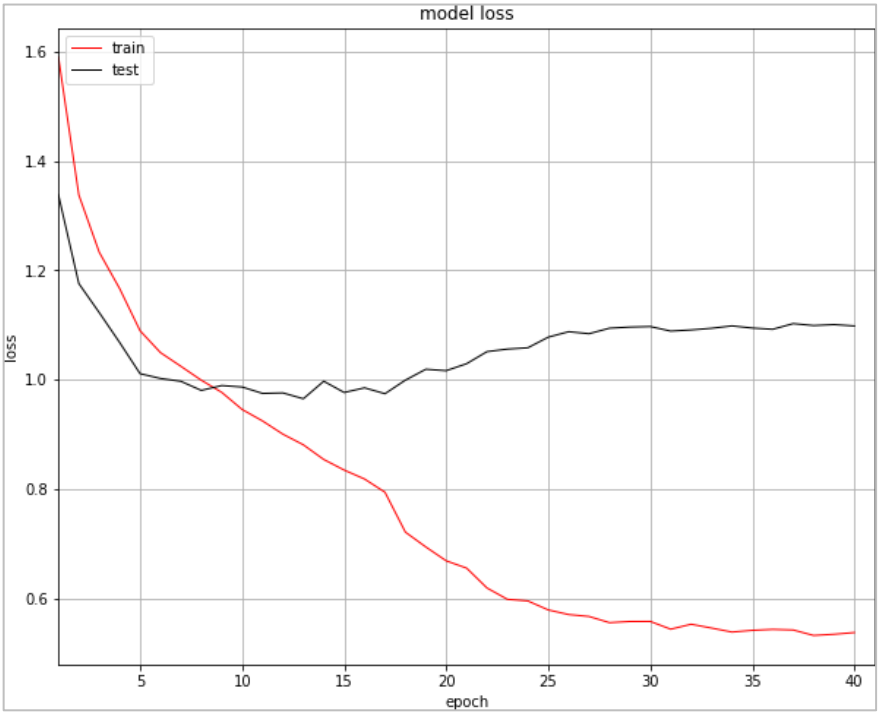


Figure 4.17. Training cross entropy loss curve of FER2013

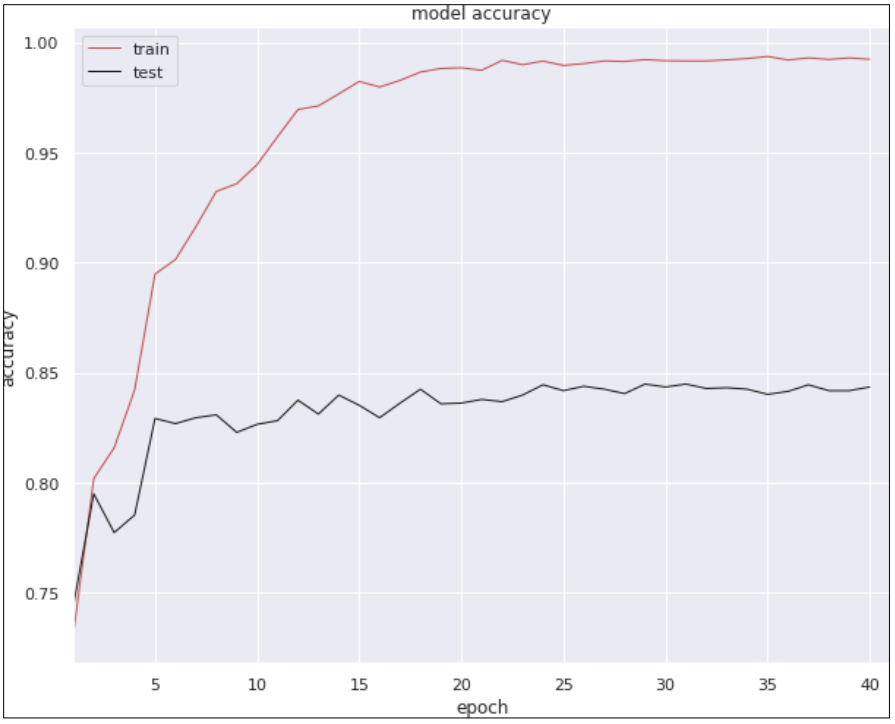


Figure 4.18. Training accuracy versus 40 training epochs on RAF-DB

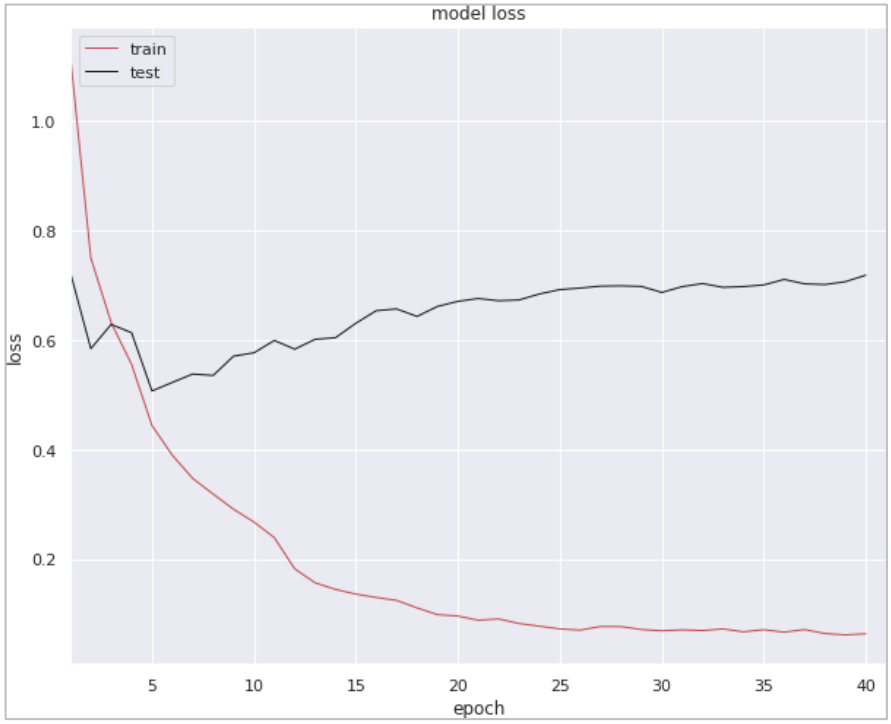


Figure 4.19. Training cross-entropy loss curve on RAF-DB

4.5. Training ResNet18 Model

Transfer learning is used to train ResNet18 pre-trained on the ImageNet dataset; we train the model on the two chosen datasets. We decided ResNet18 version was suitable for the dataset. Also, this version is capable of overfitting the training set too much, so the complexity of the model is suitable for our problem. The model is customized by replacing the top layer, which consists of 1000 neurons, with just seven neurons layer that is suitable for our number of classes of the seven basic emotions. For both datasets, we start with a learning rate of 0.0005 and use a learning rate of schedular to reduce the learning gradually until saturation.

Figures 4.20 and 4.21 show the training accuracy and cross-entropy loss on the FER2013 dataset. The training after 40 epochs stopped because the model started to overfit the training set, and validation accuracy started to decrease. The best accuracy achieved on the validation set is 68.07%.

Figures 4.22 and 4.23 show the training accuracy and cross-entropy loss of the Resnet18 model on the RAF-DB. Also, we stopped the training after 40 epochs because of the overfitting problem. The best-achieved accuracy on the validation set is 86.02%.

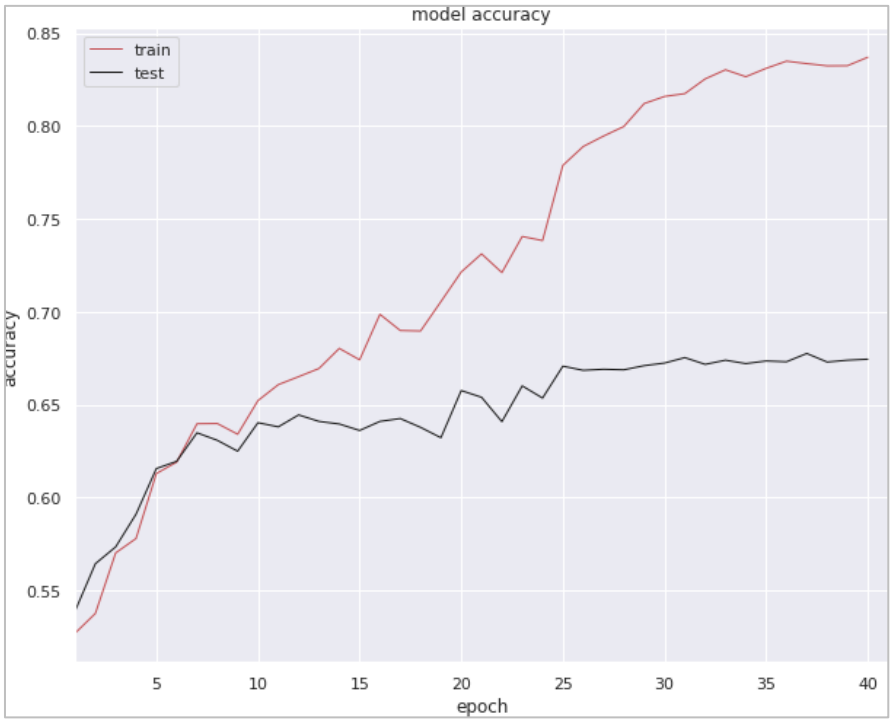


Figure 4.20. Training accuracy versus 40 training epochs on the FER2013

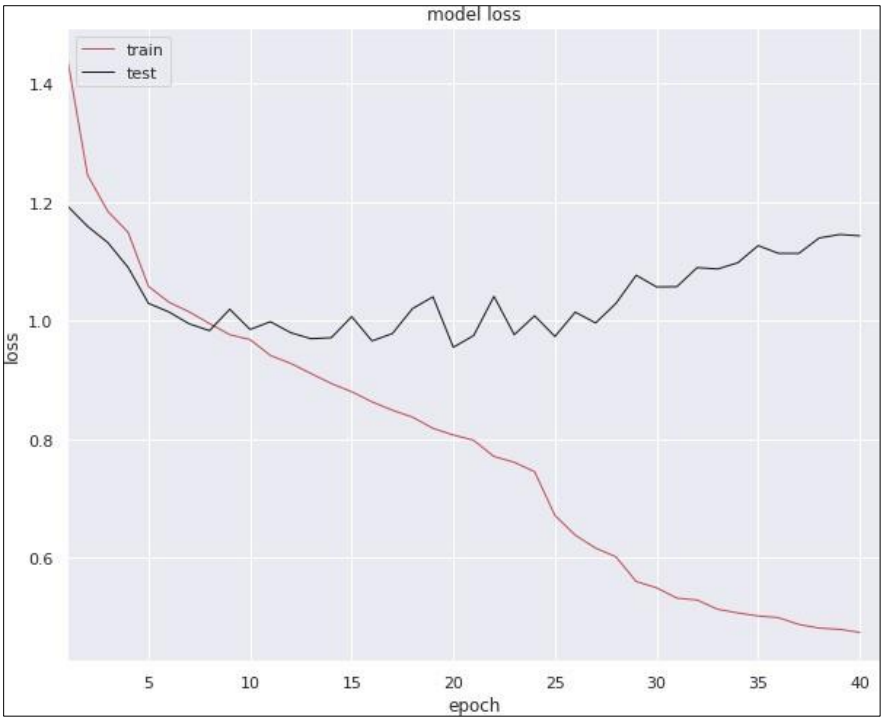


Figure 4.21. Training cross-entropy loss curve of FER2013

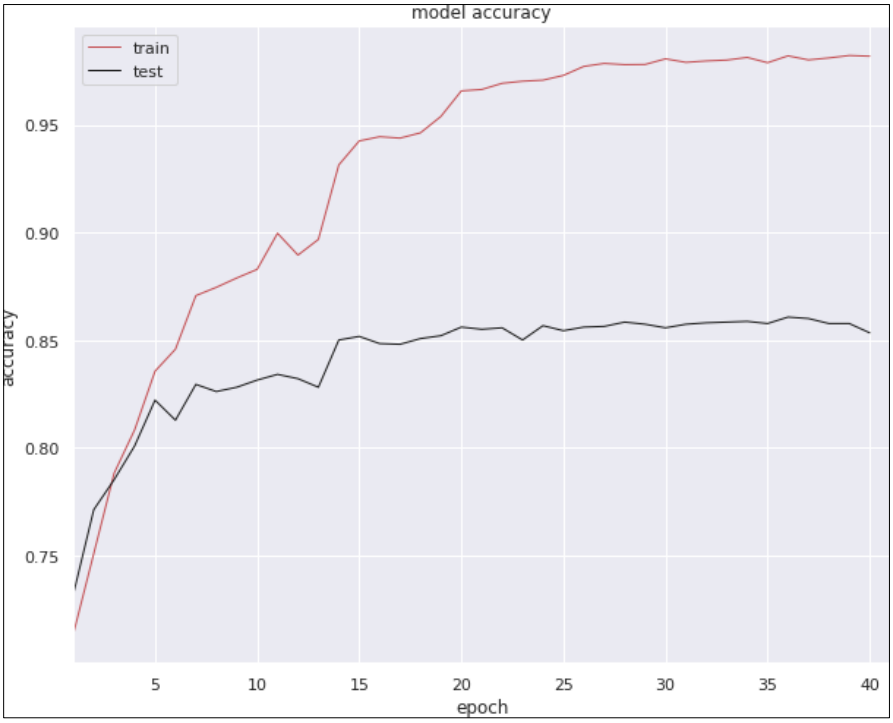


Figure 4.22. Training accuracy versus 40 training epochs on RAF-DB

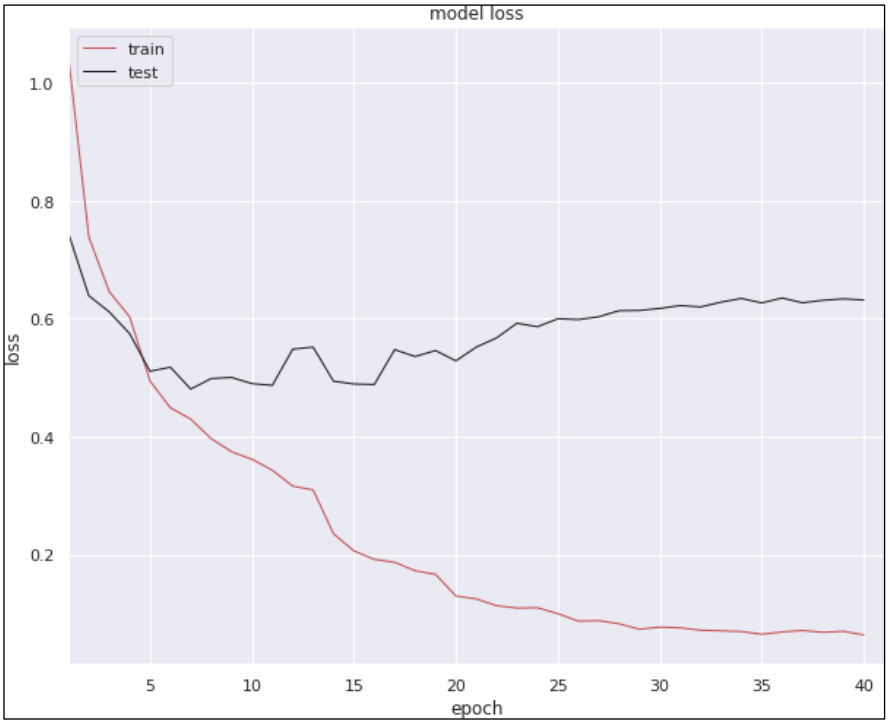


Figure 4.23. Training cross entropy loss curve on RAF-DB

4.6. Training VGGNet16 Model

The VGGNet16 model is trained on the two datasets, but we add more customization layers in this model. Firstly, we change the last average pooling layer output size from (7, 7) to (1, 1), meaning that we use global average pooling instead. Also, by changing the top layer to have 7 output neurons to represent the seven classes. This customization leads to reducing the number of parameters that exceed 138M parameters to just 33.6M parameters.

The Adam optimizer is employed with an initial learning rate of 0.0005, and a learning rate scheduler that reduces on the plateau is utilized. Additionally, the augmentation pipeline is applied as usual, similar to other algorithms that have been used. To set the parameters of the data augmentation model, an exhaustive search is conducted across different pipelines, aiming to achieve the best pipeline that provides optimal regularization results and mitigates the effect of overfitting on the training data. Figures 4.24 and 4.25 show the training accuracy of the VGGNet16 custom model. The model achieved the validation accuracy of 70.16% on the FER2013 during 40 epochs then we stopped the learning because the model started to overfit the training data too much.

Figures 4.26 and 4.27 show the training and validation of VGGNet16 model accuracy and cross-entropy loss versus 40 epochs of training time on RAF-DB. The model achieved the validation accuracy of 85.7%. We use the same parameters of the optimizer and the learning rate scheduler for both datasets and the same data augmentation pipeline.

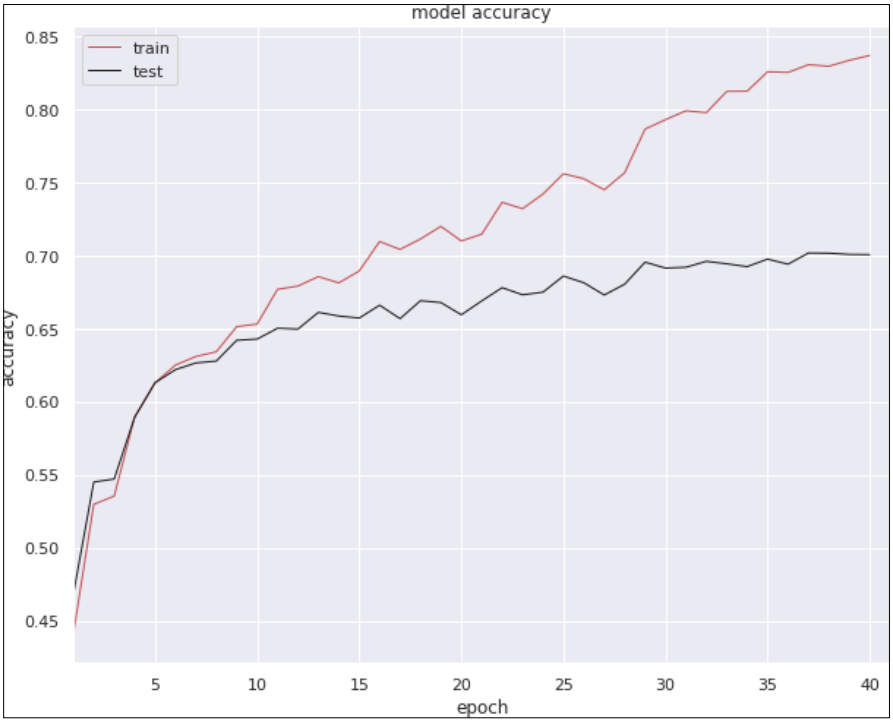


Figure 4.24. Training accuracy versus 40 training epochs on the FER2013

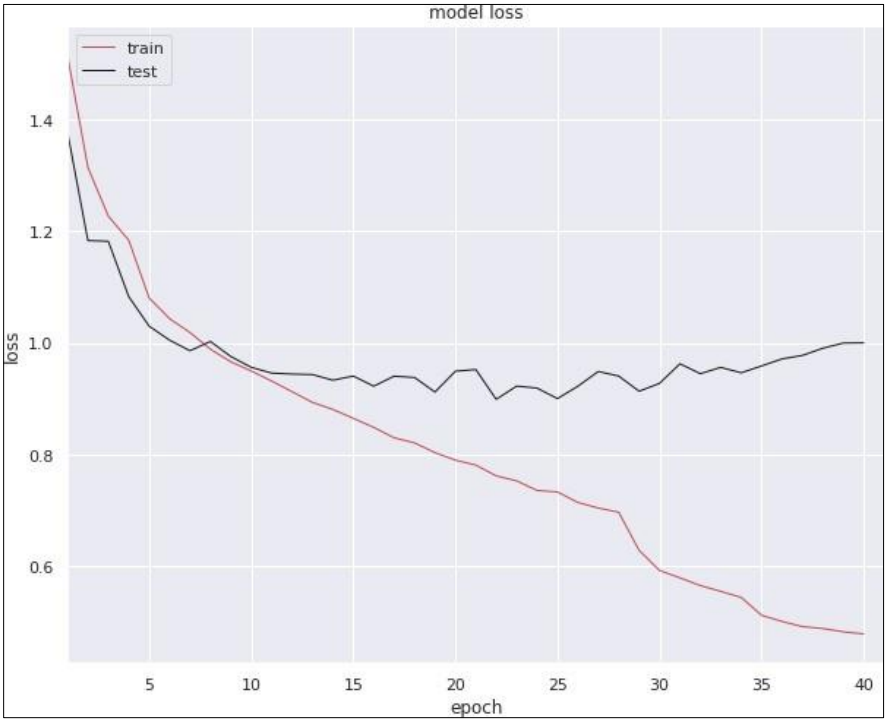


Figure 4.25. Training cross entropy loss curve of the FER2013 dataset

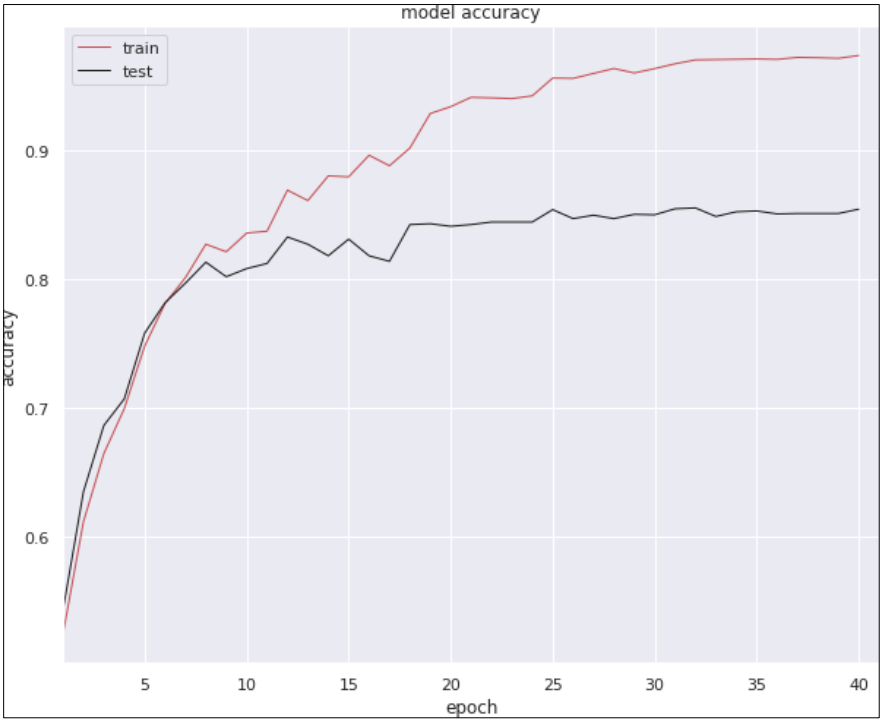


Figure 4.26. Training accuracy versus 40 training epochs on RAF-DB

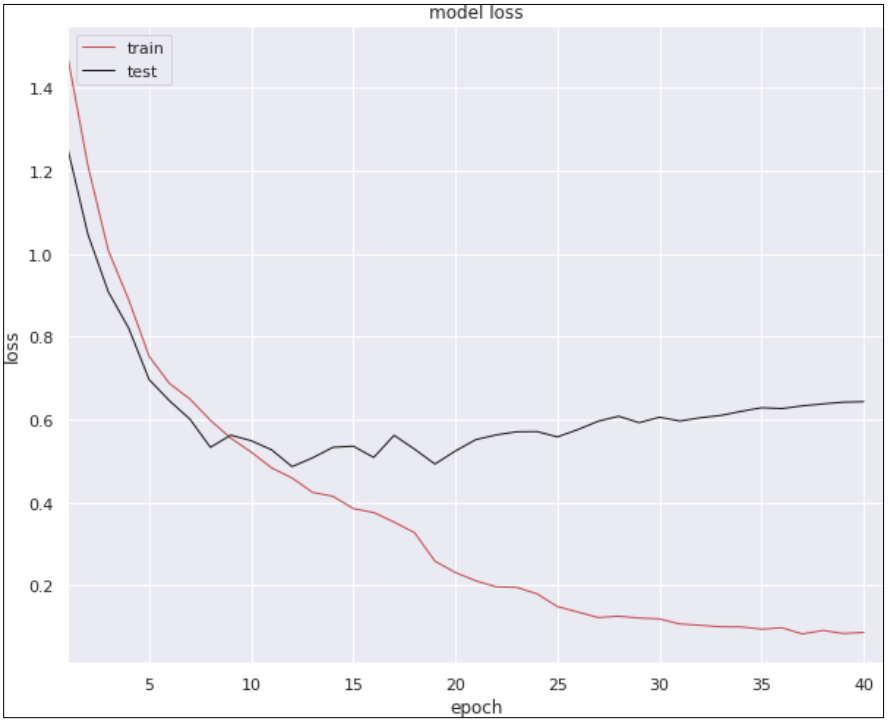


Figure 4.27. Training cross entropy loss curve on RAF-DB

4.7. Training VGGNet19 Model

A deeper and more complex data augmentation pipeline is employed to improve the performance further. The data augmentation is done using the PyTorch transforms module. Two different sets of data augmentation techniques were used - one for training and the other for testing.

The test transform applies the following operations:

- Conversion of the image to RGB,
- Resizing the image to 48x48 pixels,
- Applying TenCrop, which crops the given image into four corners and one center and then returns a list of 10 cropped images,
- Converting each cropped image to a tensor,
- Normalizing the tensor with mean and standard deviation values of [0.485, 0.456, 0.406] and [0.229, 0.224, 0.225], respectively,
- The train transform applies the following operations:
- Conversion of the image to RGB,
- Randomly cropping the image to a size of 48x48 pixels, with a scaling factor between 0.8 and 1.2,
- Applying random changes to the brightness, contrast, and saturation of the image,
- Applying random affine transformation, with a maximum translation of 20% in any direction,
- Randomly flipping the image horizontally,
- Randomly rotating the image by an angle of 45 degrees,
- Applying TenCrop, which crops the given image into four corners and one center and then returns a list of 10 cropped images,
- Converting each cropped image to a tensor,
- Normalizing the tensor with mean and standard deviation values of [0.485, 0.456, 0.406] and [0.229, 0.224, 0.225] respectively,
- Randomly erasing parts of the image with a probability of 0.5.

The VGGNet19 model used in this approach was trained on two datasets by adding more customization layers. Firstly, changing the last average pooling layer output size from (7, 7) to (1, 1) means that we use global average pooling instead. Also, the top layer is changed to have just seven output neurons to represent seven classes. This customization leads to reducing the number of parameters that exceed 138M parameters to just 45.2M parameters.

Adam optimizer is used with an initial learning rate of 0.0005, reducing the plateau learning rate scheduler. Also, the new augmentation pipeline is used. To optimally adjust the parameters of the data augmentation module and reduce the effect of overfitting the training data, the search process that tries to obtain the best pipeline among different pipelines is applied.

Figures 4.28 and 4.29 show the training accuracy of the VGGNet19 custom model. The model achieved a validation accuracy of 71% on the FER2013 during 100 epochs, and then we stopped the learning because the model started to overfit the training data too much.

Figures 4.30 and 4.31 show the training and validation of VGGNet model accuracy and cross-entropy loss versus 100 epochs training time on RAF-DB. The model achieved its best validation accuracy of 85.87%. We use the same parameters of the optimizer and the learning rate scheduler for both datasets and the same data augmentation pipeline.

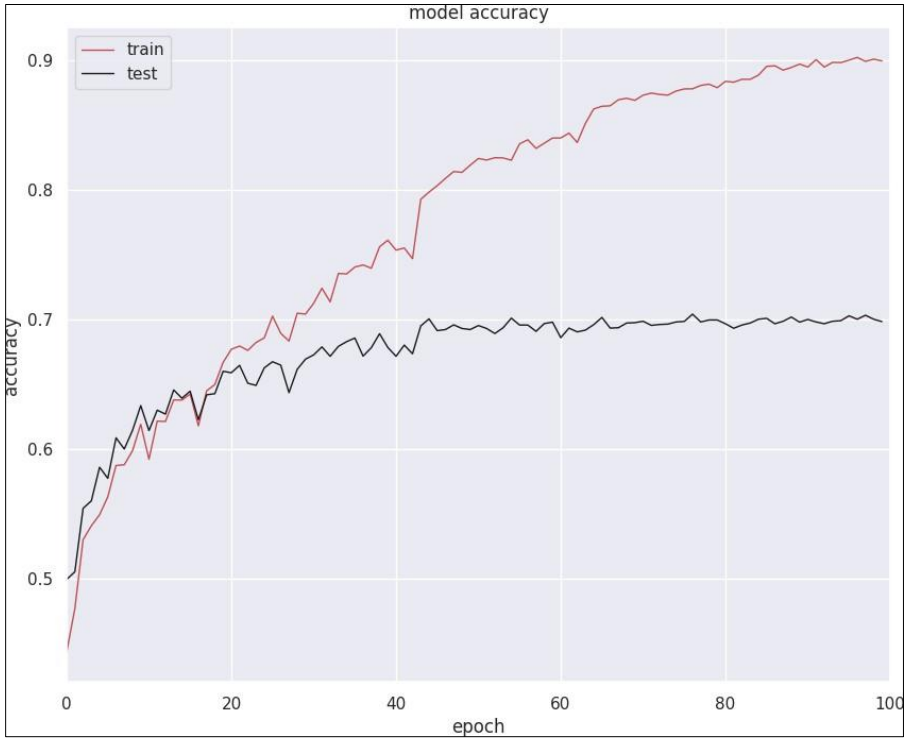


Figure 4.28. Training accuracy versus 100 training epochs on FER2013

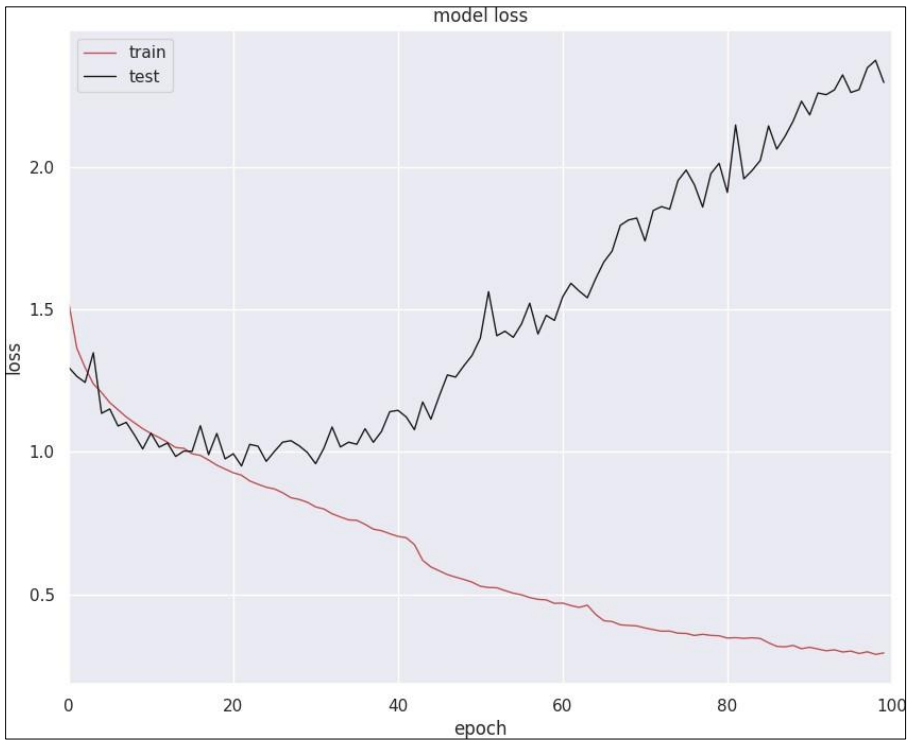


Figure 4.29. Training cross entropy loss curve of FER2013 dataset

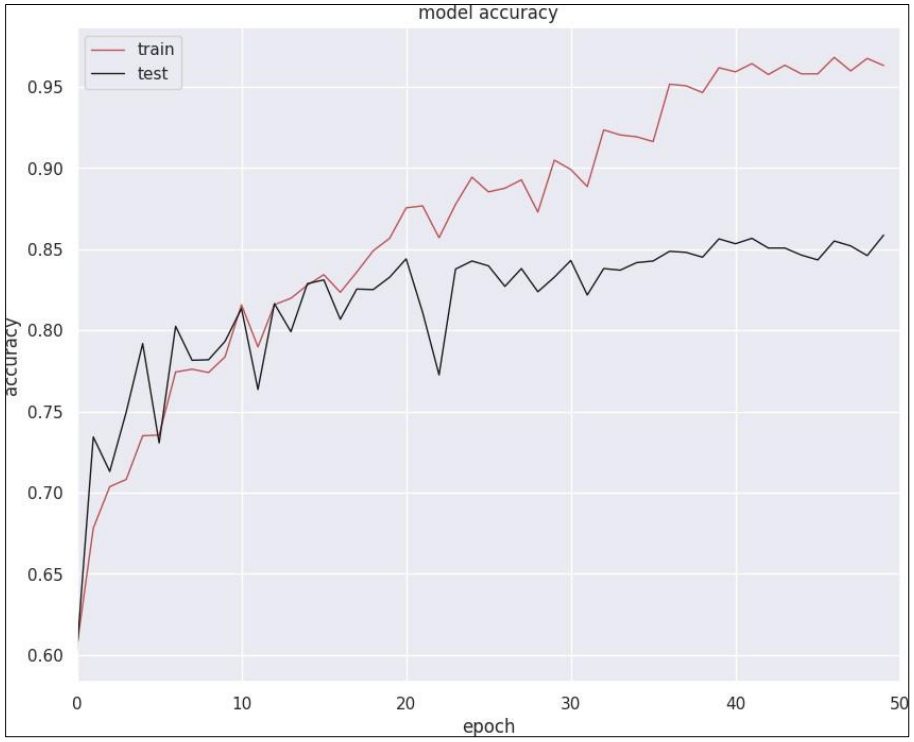


Figure 4.30. Training accuracy versus 100 training epochs on RAF-DB

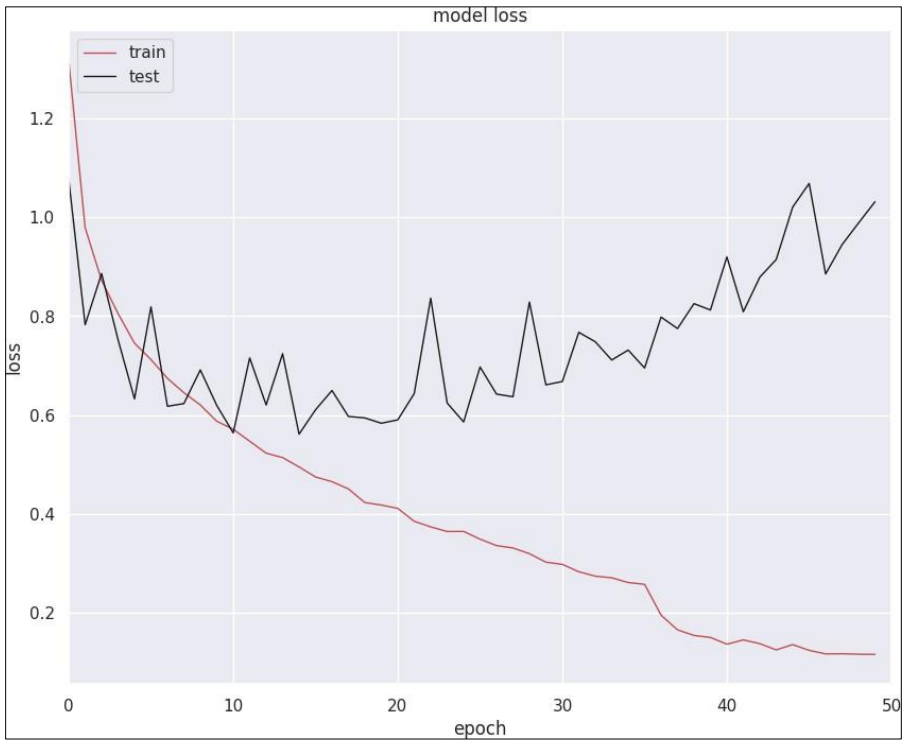


Figure 4.31. Training cross entropy loss curve on RAF-DB

4.8. Training Summary

In this chapter, the training phase of each model is presented independently. The validation set's accuracy and loss function are calculated to determine how our models perform on unseen data. Also, the features of the datasets are analysed, and some training examples related to the content of some labels in datasets are shown.

The architecture of the baseline model, the first model we implemented to measure how more complex models like ResNet18, VGGNet16, and EfficientNet models could affect the results, is discussed. Also, the genetic algorithm is employed to get the optimal model design that can fit the dataset from scratch.

For ResNet18, VGGNet16, VGGNet19, and EfficientNet models, we use transfer learning to get benefits from what these models learned from the ImageNet dataset; the use of transfer learning makes these models get above 45% accuracy after only the first epoch, which makes a great start point for learning and saves and huge learning time.

The VGGNet19 model gets the best validation accuracy on the FER2013 dataset, 71.02% accuracy and Resnet18 obtained the best validation accuracy on the RAF-DB, which brings 86% accuracy. In the next section, we will go further with the evaluation process and evaluate each model on the test set. To get more intuition about the performance of each model in detail, we also calculate the other metrics scores like precision, recall, f1-score, and confusion matrix to study the strengths and weaknesses of each model independently.

5. RESULTS AND DISCUSSION

In this chapter, the performances of each model are analysed independently, and is has been gotten more intuition about how the models perform on unseen data. The models introduced are evaluated on the test datasets presented in the previous chapter. The comparative analyses are presented under this heading.

This part briefly discusses the model's efficiency using some metrics-confusion matrices, precision per each label, recall, and f1 score. Firstly, let's start with the confusion matrix. The confusion matrix shows the types of errors the model makes and how many of them are distributed along the different labels. It’s a technique for summarizing the performance of a classification algorithm. Using classification accuracy alone can be misleading if you have an unequal number of observations in each class or if there are more than two classes in the dataset. Calculating a confusion matrix can give us a better idea of what our classification model is getting right and what types of errors it makes.

Precision, Recall, and F1-Score. These relate to getting a finer-grained idea of how well a classifier is doing, as opposed to just looking at overall accuracy. Precision is a measure of how many of the positive predictions made are correct (true positives). The recall measures how many positive cases the classifier correctly predicted over all the positive cases in the data. It is sometimes also referred to as sensitivity. F1-score is a measure combining both precision and recall. It is generally described as the harmonic mean of precision and recall. The harmonic mean is just another way to calculate an “average” of values, but it is more sensitive to the lowest values to give a more accurate metric.

5.1. Base Line Model

A baseline model is implemented from scratch to be the future work direction in the next evaluation and development and to study the effect of transfer learning in the next deep learning models. The confusion matrix of the baseline model is presented in Figure 5.1.

Also, the baseline model's precision, recall, and f1-score are calculated on each label independently in Table 5.1 to get more intuition about what the model performs on unseen data and where the labels have the most error.

This enables us to figure out the complexity of the problem we are dealing with and how it's more important to use transfer learning and test the dataset with more complex models that could memorize the data more efficiently.

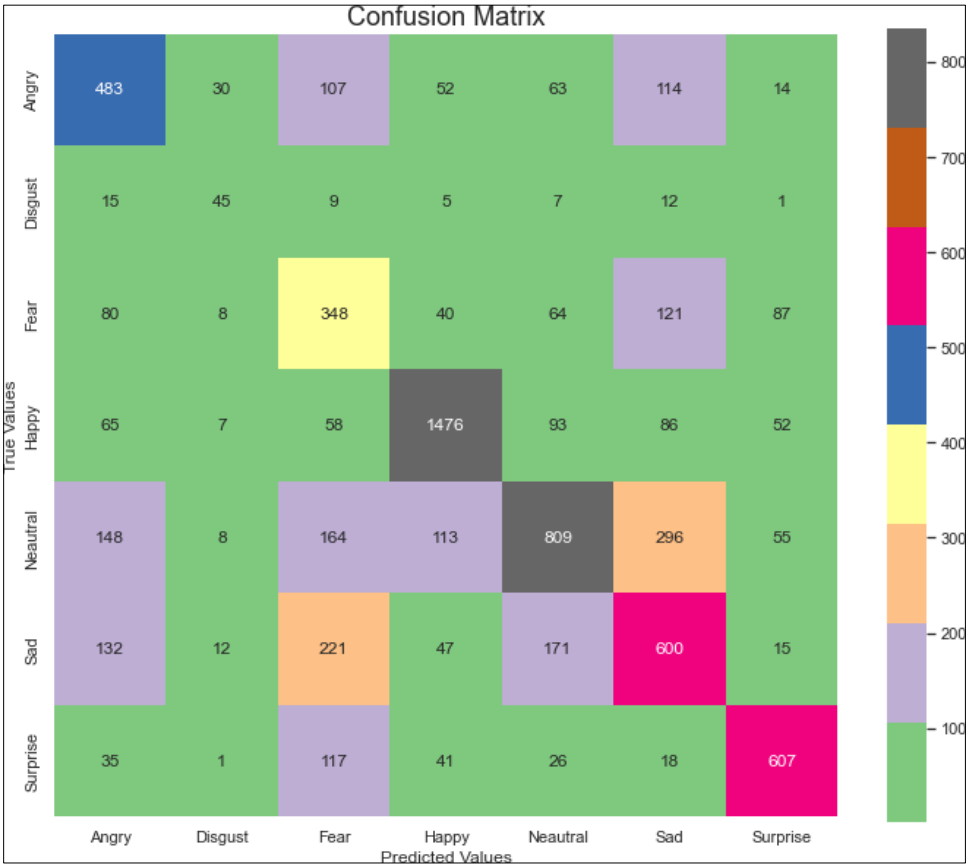


Figure 5.1. Confusion matrix result of the baseline model

Table 5.1. The classification report of the baseline model on the FER2013 shows more details about the model efficiency and metrics

Emotions	Precision	Recall	F1-score	Num. of images
Angry	0.56	0.50	0.53	958
Disgust	0.48	0.41	0.44	111
Fear	0.47	0.34	0.39	1024
Happy	0.80	0.83	0.82	1774
Neutral	0.51	0.66	0.57	1233
Sad	0.50	0.48	0.49	1247
Surprise	0.72	0.73	0.72	831
Accuracy	–	–	60.854%	7178

5.2. Genetic Algorithm

This algorithm could find a model better than the one used in Chapter 3. The Genetic algorithm increases the accuracy by 1.8% and reduces the number of model parameters from 1,672,775 to 372,423, the model is 5 times smaller than the first model.

We'll briefly go through the model efficiency using a variety of metrics. Precision for each label, recall, and f1 score for the confusion matrix. Figure 5.2 shows the confusion matrix which illustrates the various forms of errors the model produces and how many of each are divided among the various categories. It is a method for condensing the effectiveness of a classification system. Having an unbalanced number of classes can make classification accuracy alone deceptive. We can get a better understanding of what our classification is by calculating a confusion matrix. What the model is doing correctly and what kinds of mistakes it is making. Table 5.2 shows the precision, recall, and f1-score metrics evaluated over the validation dataset with total images of 7178 images.

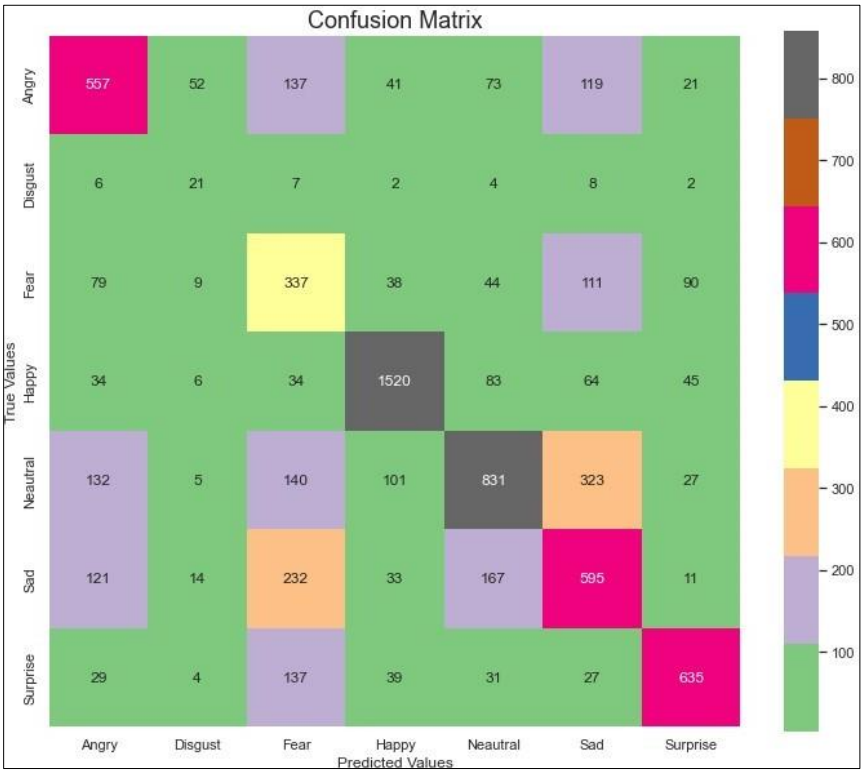


Figure 5.2. Confusion matrix of the designed genetic algorithm model

Table 5.2. The classification report of the designed genetic algorithm model of the FER2013

Emotions	Precision	Recall	F1-score	Num. of images
Angry	0.56	0.58	0.57	958
Disgust	0.42	0.19	0.26	111
Fear	0.48	0.33	0.39	1024
Happy	0.85	0.86	0.85	1774
Neutral	0.53	0.67	0.60	1233
Sad	0.50	0.48	0.49	1247
Surprise	0.72	0.76	0.73	831
Accuracy	–	–	62.598%	7178

5.3. EfficientNet Model

To get more intuition on how the Efficient model performs on the two datasets FER2013, and RAF-DB the confusion matrix, normalized confusion matrix, precision, recall, and f1-score per each label. The normalized confusion matrix is the confusion matrix divided by the number of images per label to calculate the error distribution in percent as in Figure 5.3.

Figure 5.3 shows the confusion and normalized confusion matrices of the efficient model on the FER2013 dataset. The happy and surprise labels get the best accuracy of 84% accuracy for the happy, and 79% accuracy for the surprise label. The model gets confused about 4 labels fear, angry, neutral, and sad labels which have the most error percentage.

Table 5.3 shows the classification report of the model. Calculating the precision, Recall, and F1-score per each label of the 7 basic emotions. Also, showing how many images contributed to that result gives more intuition about how our Efficient Model performs on FER2013. The surprise label gets the 0.78 f1-score even if it doesn't have more images like neutral, and sad images.

Figure 5.4 shows the confusion matrix and normalized confusion matrix of the efficient net model of the RAF-DB test set. The RAF-DB has a more robust error distribution due to the data has an RGB image with a size of 100x100 and that enables the model the capture more information about the data.

Table 5.4 shows the classification report of the model of the RAF-DB. The model achieves 84.547% accuracy on the test set. The model can predict happy images accurately with a 0.93 f1-score due to data having many happy examples that enabled the model to predict this class more accurately. Also, the three labels surprise, sad, and neutral get above 0.8 f1-score metric.

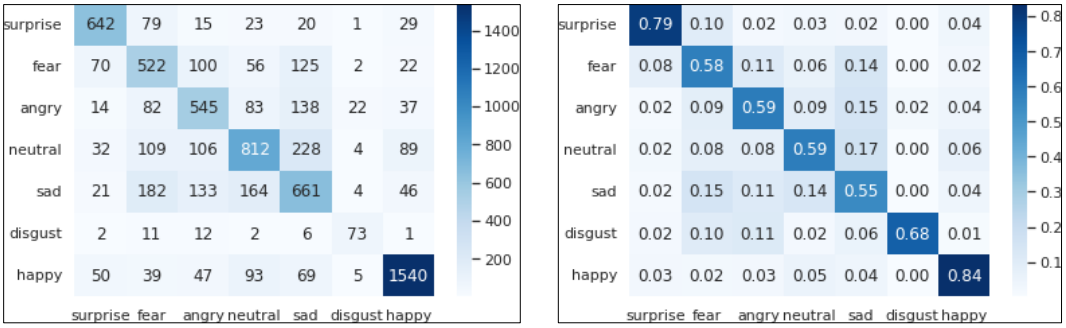


Figure 5.3. Confusion matrix and normalized confusion matrix of the EfficientNet-B0 model on the FER2013

Table 5.3. The classification report of the EfficientNet-B0 model on the FER2013 shows more details about the model efficiency and metrics

Emotions	Precision	Recall	F1-score	Num. of images
surprise	0.79	0.77	0.78	831
fear	0.58	0.51	0.54	1024
angry	0.59	0.57	0.58	958
neutral	0.59	0.66	0.62	1233
sad	0.55	0.53	0.54	1247
disgust	0.68	0.66	0.67	111
happy	0.84	0.87	0.85	1764
Accuracy	–	–	67.09%	7178

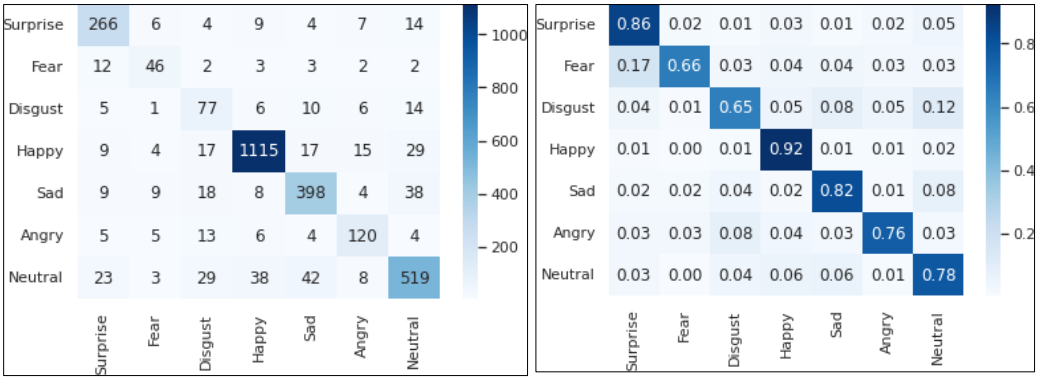


Figure 5.4. Confusion and normalized confusion matrices of EfficientNet-B0 on the RAF-DB

Table 5.4. The classification report of the EfficientNet-B0 model on the RAF-DB

Emotions	Precision	Recall	F1-score	Number of images
Surprise	0.86	0.81	0.83	329
Fear	0.66	0.62	0.64	74
Disgust	0.65	0.48	0.55	160
Happy	0.92	0.94	0.93	1185
Sad	0.82	0.83	0.83	478
Angry	0.76	0.74	0.75	162
Neutral	0.78	0.84	0.81	620
Accuracy	–	–	84.547%	3008

5.4. ResNet18

In this part, the ResNet18 model is evaluated to get more intuition about how our model performs on the test set. For each dataset, the confusion matrix and normalized confusion matrix are computed, also the classification report is presented that shows the precision, recall, and f1 score per each label of our 7 basic emotions. Table 5.5 shows the classification report of the model on the FER2013 dataset the model achieved a total accuracy of 68.07% on the test set, Figure 5.5 show the confusion matrix and normalized confusion matrix of the ResNet18 model on the FER2013 test set.

The last two figures show the confusion matrix and normalized confusion matrix of the ResNet18 model on the RAF-DB test set. We can easily observe that the model is more robust on this dataset, also it could reach a reasonable accuracy compared to other studies in the literature.

Table 5.6 shows more details about how our model performs on the RAF-DB test set, it

shows the precision, recall, and f1-score per each label of our seven basic emotions. The model achieved a test accuracy of 86.02%. The model gets a 0.94 f1 score on the happy label, and also it gets above 0.80 f1-scores on 4 labels, surprise, sad, angry, and neutral label.

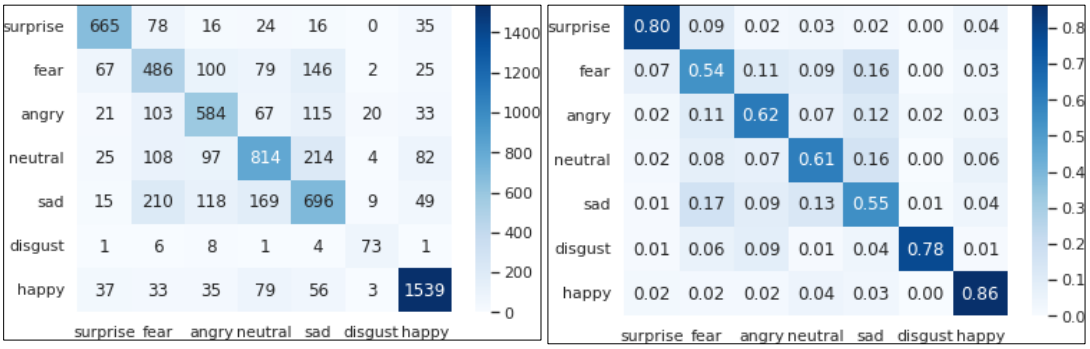


Figure 5.5. Confusion and normalized confusion matrices of ResNet18 on the FER2013

Table 5.5. Classification report of the ResNet18 model on the FER2013

Label	Precision	Recall	F1-score	Num. of images
surprise	0.80	0.80	0.80	831
fear	0.54	0.47	0.50	1024
angry	0.62	0.61	0.61	958
neutral	0.61	0.66	0.63	1233
sad	0.55	0.56	0.55	1247
disgust	0.78	0.66	0.71	111
happy	0.86	0.87	0.87	1764
Accuracy	–	–	68.07%	7178

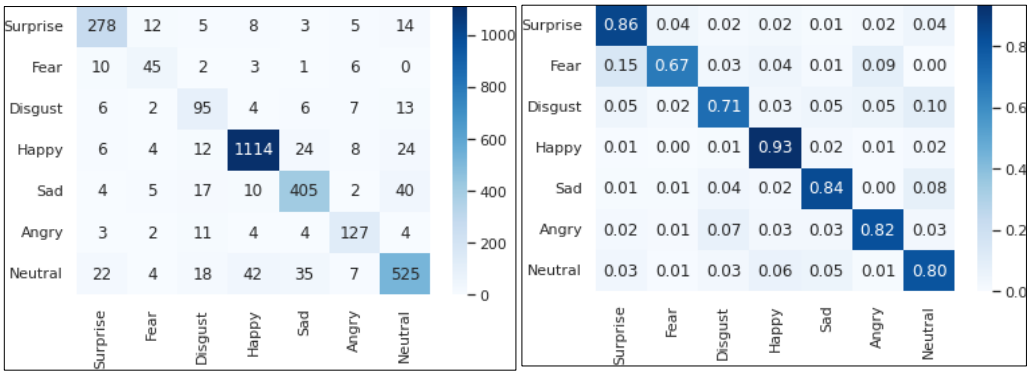


Figure 5.6. Confusion and normalized confusion matrices of ResNet18 on the RAF-DB

Table 5.6. Classification report of the ResNet18 model on the RAF-DB

Label	Precision	Recall	F1-score	Num. of images
Surprise	0.86	0.84	0.85	329
Fear	0.67	0.61	0.64	74
Disgust	0.71	0.59	0.65	160
Happy	0.93	0.94	0.94	1185
Sad	0.84	0.85	0.84	478
Angry	0.82	0.78	0.80	162
Neutral	0.80	0.85	0.82	620
Accuracy	–	–	86.02%	3008

5.5. VGGNet16 Model

The VGGNet16 model is evaluated on the test set and the confusion matrix and normalized confusion matrix to get more intuition about how our model performs in unseen data. The VGGNet16 model achieves the 70.2% test accuracy on the FER2013. The original model of VGGNet16 has 138M parameters and this number is too big compared with other models and the complexity is too high but the customization we added reduced the number of parameters to 33.6M parameters, which is more reasonable.

It has been given the confusion matrix and the normalized confusion matrix of the model on the FER2013 test set in Figure 5.7. The VGGNet16 model reduced the error on all labels compared with another model we can notice that by comparing the diagonal of the VGGNet16 confusion matrix with all other model's confusion matrices.

Table 5.7 shows the precision, recall, and f1 score of all labels, and how many images per label contributed to calculating these metrics. From the table, the f1 score of each label is

increased and the overall accuracy on the test set is 70.2%. We also show the confusion matrix and normalized confusion matrix of the VGGNet16 model on the RAF-DB, in Figure 5.8. The ResNet18 model achieved the best result on this dataset, but the VGGNet16 model gets 85.88% on the test set, while ResNet18 gets 86.02% difference in accuracy was not great but it is important to note that the VGGNet16 model has more parameters and its complexity is too high. Table 5.8 shows more details about how the VGGNet16 model performs on each label.

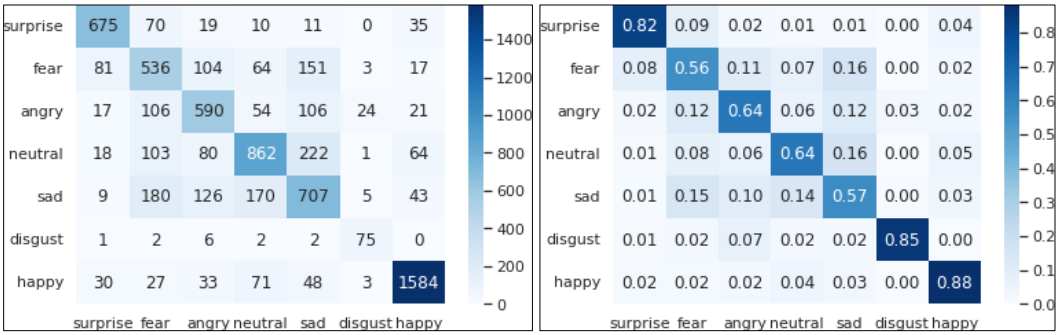


Figure 5.7. Confusion and normalized confusion matrices of VGGNet16 on the FER2013

Table 5.7. The classification report of the VGGNet16 model on the FER2013

Emotions	Precision	Recall	F1-score	Num. of images
surprise	0.82	0.81	0.82	831
fear	0.56	0.52	0.54	1024
angry	0.64	0.62	0.63	958
neutral	0.64	0.70	0.67	1233
sad	0.57	0.57	0.57	1247
disgust	0.85	0.68	0.75	111
happy	0.88	0.90	0.89	1764
Accuracy	–	–	70.2%	7178

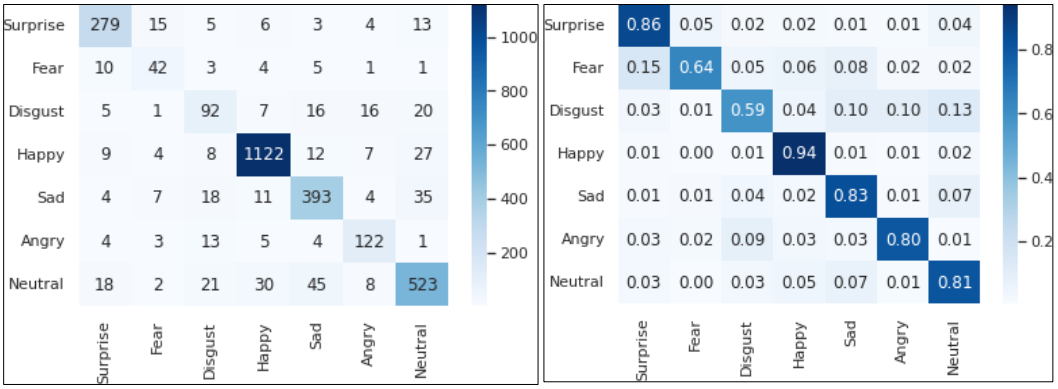


Figure 5.8. Confusion and normalized confusion matrices of VGGNet16 on the RAF-DB

Table 5.8. The classification report of the VGGNet16 model on the RAF-DB

Emotions	Precision	Recall	F1-score	Num. of images
Surprise	0.86	0.85	0.85	329
Fear	0.64	0.57	0.60	74
Disgust	0.59	0.57	0.58	160
Happy	0.94	0.95	0.95	1185
Sad	0.83	0.82	0.83	478
Angry	0.80	0.75	0.78	162
Neutral	0.81	0.84	0.83	620
Accuracy	–	–	85.88%	3008

5.6. VGGNet19

The VGGNet19 model was evaluated on the test set and the confusion matrix and normalized confusion matrix to get more intuition about how our model performs in unseen data. The VGGNet19 model achieved the best result on the FER2013 dataset, it gets a 71.2% test accuracy. VGGNet19's original model has 138M parameters, and this number is too big compared with other models, and the complexity is too high, but the customization we add reduces the number of parameters to 40.2M parameters, and this is more reasonable.

Figure 5.9 shows the confusion matrix and the normalized confusion matrix of the model on the FER2013 test set. The VGGNet19 model reduced the error on all labels to another model we can notice that by comparing the diagonal of the VGGNet19 confusion matrix with all other model’s confusion matrices.

Table 5.9 shows the precision, recall, and f1 score metrics of all labels and how many images per label contribute to calculating these metrics. From the table, we can notice that the f1

score of each label is increased, and the overall accuracy on the test set is 71.02%. We also show the confusion matrix and normalized confusion matrix of the VGGNet model on the RAF-DB in Figure 5.10. The ResNet18 model gets the best result on this dataset, but the VGGNet19 model gets 85.87% on the test set, and ResNet18 gets 86.02%, which is not a big difference in accuracy, but the VGGNet19 model has more parameters, and its complexity is too big. Table 5.10 shows more details about how the VGGNet model performs on each label to get more intuition about how the model performs on unseen data.

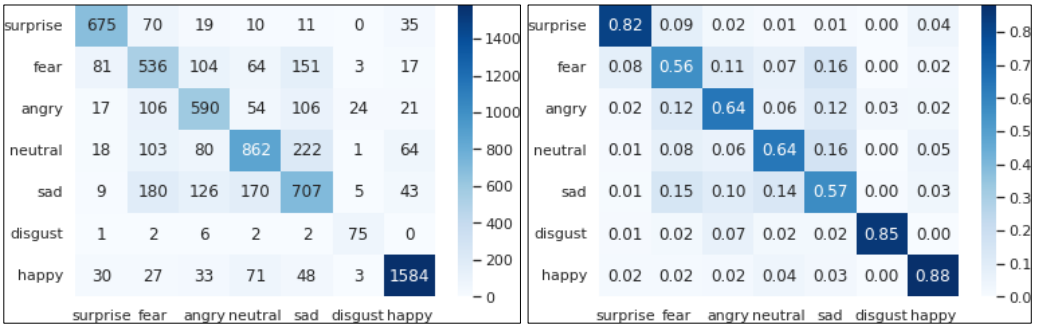


Figure 5.9. Confusion and normalized confusion matrices of VGGNet19 on the FER2013

Table 5.9. The classification report of the VGGNet19 model on the FER2013

Emotions	Precision	Recall	F1-score	Num. of images
surprise	0.84	0.79	0.82	831
fear	0.57	0.59	0.58	1024
angry	0.61	0.64	0.64	958
neutral	0.68	0.67	0.64	1233
sad	0.59	0.57	0.58	1247
disgust	0.69	0.72	0.85	111
happy	0.89	0.90	0.89	1764
Accuracy	–	–	71.02%	7178

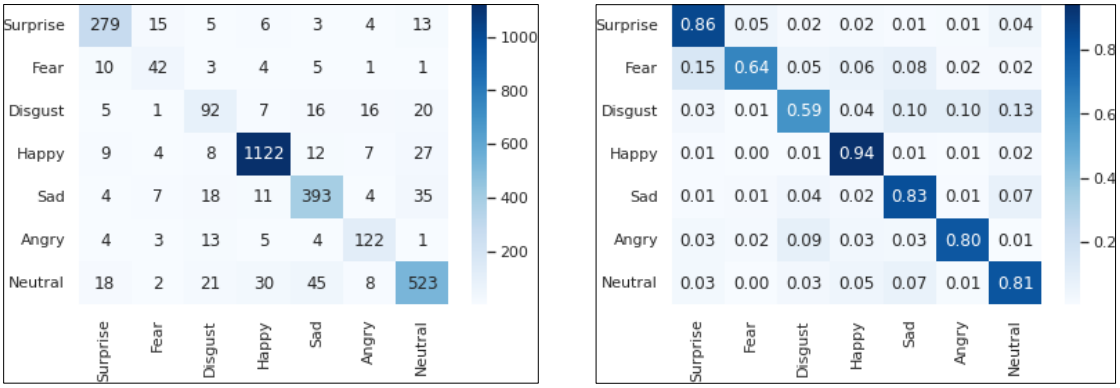


Figure 5.10. Confusion and normalized confusion matrices of VGGNet19 on the RAF-DB

Table 5.10. The classification report of the VGGNet19 model on the RAF-DB

Emotions	Precision	Recall	F1-score	Num. of images
Surprise	0.87	0.83	0.85	329
Fear	0.77	0.50	0.61	74
Disgust	0.66	0.62	0.64	160
Happy	0.92	0.95	0.94	1185
Sad	0.84	0.83	0.83	478
Angry	0.87	0.77	0.82	162
Neutral	0.81	0.85	0.83	620
Accuracy	–	–	85.87%	3008

5.7. The Comparison of Models

This sub-chapter shows the results of all models and compares the results based on test accuracy. Six different models are applied to the FER2013 dataset, firstly by applying the baseline model from scratch, and then by using the genetic algorithm to try to design a more robust model. After that, four different Deep Convolution neural network architectures are employed, ResNet18, EfficientNet-B0, VGGNet16, and VGGNet19. The accuracy of these deep learning approaches is more promising and was further validated RAF-DB dataset. Table 5.11 shows the overall results of all models on the FER2013 dataset, the Deep Learning model gets the best accuracy compared to the genetic algorithm, and baseline model. VGGNet19 achieved the highest accuracy 71.02% for FER2013. However, the number of model parameters and the complexity of the model should be taken into consideration.

Table 5.12 shows the overall results on RAF-DB, based on the FER2013 results we find that

the three deep learning models are promising for solving the problem of emotion recognition therefore, we only used these deep learning models on the RAF-DB. For this dataset, the ResNet18 model gets the best results with a test accuracy of 86.02%.

Table 5.11. All model results on the FER2013 test set

Model	Test Accuracy	Number of Parameters
Base Line	60.854%	1,672,775
GA Model	62.598%	372,423
EfficientNet-B0	67.09%	4,016,515
ResNet18	68.07%	11,180,103
VGGNet16	70.20%	33,625,927
VGGNet19	*71.02%*	*45,227,079*

Table 5.12. All model results on RAF-DB test set

Model	Test Accuracy	Number of Parameters
EfficientNet-B0	84.55%	4,016,515
VGGNet16	85.88%	33,625,927
VGGNet19	85.87%	45,227,079
ResNet18	*86.02%*	*11,180,103*

5.8. Study robustness

To ensure the approach is robust the proposed pipeline is employed for two datasets, and we not only use one Algorithm for solving the problem, but we try different approaches and models to try to get robust models that represent our work. To get the model's hyperparameters and training augmentation pipeline, we performed an exhaustive search between various pipelines, to find robust hyperparameters that give us the best result representative of each model independently. We ensured the robustness of our approach by applying our pipeline to two open sources of datasets. By comparing our results with other researchers, we get a better performance model and achieve the best results on those datasets by tackling the problem with many models and different hyperparameters. Tables 5.13 and 5.14 compare our best results with different state-of-the-art models that have been previously applied to both datasets. The accuracy of VGGNet19 on FER2013 and ResNet18 on RAF-DB outperform all other models. Some researchers depend on multiple datasets to merge them and construct one bigger dataset; however, in our study we relied only on each dataset independently without merging. The customization layers that used for each model decreased the number of parameters of the models, and the inference time also increased compared to with the original published models.

Tablet 5.13. Comparison of the results on the FER2013

Study	Accuracy for test set
Devries et al.	67.21%
Zhang et al.	70.60%
Ours (VGGNet19)	71.02%

Tablet 5.14. Comparison of the results on the RAF-DB

Model	Accuracy for test set
gACNN	85.07%
DLP-CNN	74.20%
Ours (ResNet18)	86.02%

5.9. Testing Summary

This chapter shows the results of each model independently and got more deeply into the model's results by using different metrics functions. The strengths and weaknesses of each model are highlighted independently, and then a brief comparison of all these models with each other is presented to give brief discussions and comparisons between all these models. These results prove that CNN architectures like ResNet, EfficientNet, and VGGNet could achieve great results on the Emotion recognition problem and should be considered for future improvement in this work. The VGGNet19 model outperforms other models on the FER2013 dataset, achieving a test accuracy of 71.02%. The original VGGNet model is excessively intricate and voluminous due to its 138 million parameters, making it more extensive and more complex than other models. However, the customization has been implemented to reduce the parameter count to 45.2 million, which is more reasonable. Nevertheless, VGGNet19 achieves the test accuracy of 85.87%, while ResNet18 achieves the test accuracy of 86.02%.

6. CONCLUSION AND FUTURE DIRECTIONS

6.1. Outline of the Contribution

Facial expressions can be studied to better understand human emotions and intentions since they provide much nonverbal information. The strategy outlined in this study demonstrated success in facial expression recognition performance and offered a fresh perspective on issues previously raised in the literature. Our novel method makes it possible to recognize and classify facial expressions more accurately and effectively, which lowers computing costs and time requirements while increasing image recognition rates. The model is created to improve the categorization accuracy of face images. Our results indicate that deep learning-enabled facial expression recognition methods increase productivity through enhanced facial identification, accuracy, and feature and expression interpretation.

Facial expression emotion identification is an intriguing subject of study with applications in many domains, including safety, health, and human-machine interactions. By creating techniques for reading, coding, and extracting facial expressions, researchers in this field are attempting to enhance computer predictions. The extraordinary success of deep learning has led to the deployment of many types of architectures to boost performance.

6.2. Overall Conclusion

The CNN is employed on the FER2013 dataset and studies the effect of the residual learning blocks to increase the model's performance by choosing the number of layers and neurons based on our intuition of applying CNN for different datasets. The trial-and-error technique has been used to pick the best learning rate and scheduler.

Another technique for evolving the CNN architecture using a Genetic Algorithm (GA) is explored. A given CNN that is utilized to handle image classification tasks is improved through the usage of the GA by showing the parameters that need to be tuned.

The GA approach is employed to get a better model than the baseline model that can fit the FER2013 dataset more wisely, increased the accuracy by 1.8%, and reduced the model's size five times the baseline model.

Our proposed model can be combined with other models to increase the accuracy of face recognition systems utilizing the FER2013 dataset because it is computationally less expensive. Although the FER2013 dataset is very complicated with a small number of samples per class, the number of samples in each class can be increased by the right amount to improve accuracy. To solve this problem, transfer learning techniques have been used to re-train VGGNet16, VGGNet19, ResNet18, and EfficientNet-B0 on two open sources of datasets, FER2013 and RAF-DB. Using pre-trained deep CNN architectures proved an outstanding performance on these datasets.

The proposed model is evaluated on the test set and generates the confusion matrix and normalized confusion matrix to understand better how our model functions when dealing with unobserved input.

The VGGNet19 model performs the best on the FER2013 dataset, with a test accuracy of 71.02%. In contrast to other models, more customization layers are added to this model. Firstly, we change the last average pooling layer output size from (7, 7) to (1, 1), meaning we use global average pooling instead. Also, by changing the top layer just to have seven output neurons to represent seven classes. The VGGNet19 original model has 138M parameters, making it too complicated and huge. However, thanks to the customization, the number of parameters is reduced to 45.2M, which is more reasonable. Unlike the tests on the other dataset, the ResNet18 model outperformed all other models with an accuracy of 86.02% in the experiments on the RAF-DB dataset.

Enhancing facial expression recognition systems by integrating natural language processing (NLP) can add depth to their capabilities. In the future, this approach could have substantial implications for e-health systems and healthcare services. Combining linguistic context with facial emotions can significantly improve emotion prediction accuracy, offering richer information for more precise motion prediction.

REFERENCES

1. Mnih, V., and Hinton, G. E. (2010). *Learning to detect roads in high-resolution aerial images*. In Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part VI 11. Springer Berlin Heidelberg, pp. 210-223

2. Mnih, V., and Hinton, G. E. (2012). *Learning to label aerial images from noisy data*. In Proceedings of the 29th International conference on machine learning (ICML-12), pp. 567-574.

3. Zhang, Z., Wang, Y., Liu, Q., Li, L., and Wang, P. (2016, July). *A CNN based functional zone classification method for aerial images*. In 2016 IEEE international geoscience and remote sensing symposium (IGARSS). IEEE pp. 5449-5452.

4. Rosenstein, D., and Oster, H. (1988). *Differential facial responses to four basic tastes in newborns. Child development,* 1555-1568.

5. Pease, A. (1997). *How to Read Others' Thoughts by Their Gestures*. Sheldon.

6. Ekman, P., Friesen, W. V., O'sullivan, M., Chan, A., Diacoyanni-Tarlatzis, I., Heider, K., and Tzavaras, A. (1987). Universals and cultural differences in the judgments of facial expressions of emotion. *Journal of personality and social psychology*, *53*(4), 712.

7. Pease, B., and Pease, A. (2008). *The definitive book of body language: The hidden meaning behind people's gestures and expressions*. Bantam. Pease international.

8. Gadarian, S. K., and Brader, T. (2023). *Emotion and Political Psychology*. The Oxford Handbook of Political Psychology, 191.

9. So, W. C., Demir, Ö. E., and Goldin-Meadow, S. (2010). When speech is ambiguous, gesture steps in: Sensitivity to discourse-pragmatic principles in early childhood. *Applied psycholinguistics*, *31*(1), 209-224.

10. Internet: *Gesture and environment*, King's Crown Press., Morningside Heights, New York (1941), Url: https://psycnet.apa.org/record/1942-00254-000 . Last accessed date 11/12/2023.

11. Kendon, A. (1981). *The study of gesture: Some remarks on its history*. In Semiotics. Boston, MA: Springer US,pp. 153-164.

12. Gunes, H., and Piccardi, M. (2005, August). *Fusing face and body gesture for machine recognition of emotions*. In ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, *2005.* IEEE. pp. 306-311.

13. Gunes, H., and Piccardi, M. (2006, August). *A bimodal face and body gesture database for automatic analysis of human nonverbal affective behavior*. In 18th International conference on pattern recognition (ICPR'06). IEEE Vol. 1, pp. 1148-1153.

14. Gunes, H., Shan, C., Chen, S., and Tian, Y. (2015). Bodily expression for automatic affect recognition. *Emotion recognition: A pattern analysis approach*, 343-377.

15. Internet: *FER2013 dataset*, url: https://www.kaggle.com/datasets/msambare/fer2013, access date: December 1, 2023. Last accessed date 12/12/2023 .

16. Goodfellow, I. J., Erhan, D., Carrier, P. L., Courville, A., Mirza, M., Hamner, B., and Bengio, Y. (2013). *Challenges in representation learning: A report on three machine learning contests*. In Neural Information Processing: 20th International Conference, ICONIP 2013, Daegu, Korea. Proceedings, Part III 20. Springer berlin Heidelberg,pp. 117-124.

17. Internet: *RAF-DB*, url: https://www.kaggle.com/datasets/shuvoalok/raf-db-dataset. Last access date: 01/12/2023.

18. He, K., Zhang, X., Ren, S., and Sun, J. (2016). *Deep residual learning for image recognition*. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778.

19. Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:*1409.1556.

20. Tan, M., and Le, Q. (2019, May). *EfficientNet: Rethinking model scaling for convolutional neural networks*. In International conference on machine learning. PMLR, pp. 6105-6114.

21. Internet: LeCun, Y. (1998). *The MNIST database of handwritten digits*. Url: http://yann. lecun. com/exdb/mnist/. Last access date 12/12/2023.

22. Heravi, E. J., Aghdam, H. H., and Puig, D. (2016, September). Classification of Foods Using Spatial Pyramid Convolutional Neural Network. In *CCIA*,pp. 163-168.

23. Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84-90.

24. Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

25. Kollias, D., Sharmanska, V., and Zafeiriou, S. (2019). Face behavior a la carte: Expressions, affect and action units in a single network. *arXiv preprint arXiv:*1910.11111.

26. Kollias, D., and Zafeiriou, S. (2019). Expression, affect, action unit recognition: Aff-wild2, multi-task learning and arcface. *arXiv preprint arXiv:1910.04855*.

27. Kollias, D., Sharmanska, V., and Zafeiriou, S. (2021). Distribution matching for heterogeneous multi-task learning: a large-scale face study. *arXiv preprint arXiv:*2105.03790.

28. Kollias, D., and Zafeiriou, S. (2021). Affect analysis in-the-wild: Valence-arousal, expressions, action units and a unified framework. *arXiv preprint arXiv:*2103.15792.

29. Kollias, D., Schulc, A., Hajiyev, E., and Zafeiriou, S. (2020, November). *Analysing affective behavior* in the first abaw competition. In 2020 15th IEEE International Conference on Automatic Face and Gesture Recognition. IEEE,pp. 637-643.

30. Mollahosseini, A., Hasani, B., & Mahoor, M. H. (2017). AffectNet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, *10*(1), 18-31.

31. Ringeval, F., Sonderegger, A., Sauer, J., and Lalanne, D. (2013, April). *Introducing the RECOLA multimodal corpus of remote collaborative and affective interactions*. In 2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG). IEEE ,pp. 1-8.

32. Dhall, A., Goecke, R., Ghosh, S., Joshi, J., Hoey, J., and Gedeon, T. (2017, November). *From individual to group-level emotion recognition: Emotiw 5.0*. In Proceedings of the 19th ACM international conference on multimodal interaction, pp. 524-528.

33. Kossaifi, J., Tzimiropoulos, G., Todorovic, S., and Pantic, M. (2017). AFEW-VA database for valence and arousal estimation in-the-wild. *Image and Vision Computing*, 65, 23-36.

34. Jorge-Martinez, D., Butt, S. A., Onyema, E. M., Chakraborty, C., Shaheen, Q., De-La-Hoz-Franco, E., and Ariza-Colpas, P. (2021). Artificial intelligence-based Kubernetes container for scheduling nodes of energy composition. *International Journal of System Assurance Engineering and Management*, 1-9.

35. Afriyie, R., Asante, M., and Onyema, E. M. (2020). Implementing morpheme-based compression security mechanism in distributed systems. *International Journal of Innovative Research & Development (IJIRD), 9*(2), 157-162.

36. Ahirwar, M. K., Shukla, P. K., and Singhai, R. (2021). CBO-IE: a data mining approach for healthcare IoT dataset using chaotic biogeography-based optimization and information entropy. *Scientific Programming*, 2021, 1-14.

37. Bhatt, R., Maheshwary, P., Shukla, P., Shukla, P., Shrivastava, M., and Changlani, S. (2020). Implementation of fruit fly optimization algorithm (FFOA) to escalate the attacking efficiency of node capture attack in wireless sensor networks (WSN). *Computer Communications*, 149, 134-145.

38. Zhang, Z., Luo, P., Loy, C. C., and Tang, X. (2015). *Learning social relation traits from face images*. In Proceedings of the IEEE International Conference on Computer Vision, pp. 3631-3639.

39. Devries, T., Biswaranjan, K., and Taylor, G. W. (2014, May). *Multi-task learning of facial landmarks and expression*. In 2014 Canadian conference on computer and robot vision. IEEE ,pp. 98-103.

40. Li, S., Deng, W., and Du, J. (2017). *Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild*. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2852-2861.

41. Li, Y., Zeng, J., Shan, S., and Chen, X. (2018). *Occlusion aware facial expression recognition using CNN with attention mechanism*. IEEE Transactions on Image Processing, *28*(5), 2439-2450.

42. Wang, K., Peng, X., Yang, J., Meng, D., and Qiao, Y. (2020). *Region attention networks for pose and occlusion robust facial expression recognition*. IEEE Transactions on Image Processing, 29, 4057-4069.

43. Chang, W. Y., Hsu, S. H., and Chien, J. H. (2017). *FATAUVA-Net: An integrated deep learning framework for facial attribute recognition, action unit detection, and valence-arousal estimation*. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 17-25.

44. Li, J., Chen, Y., Xiao, S., Zhao, J., Roy, S., Feng, J., and Sim, T. (2017). *Estimation of affective level in the wild with multiple memory networks*. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 1-8.

45. Hasani, B., and Mahoor, M. H. (2017). *Facial affect estimation in the wild using deep residual and convolutional networks*. In *Proceedings of the* IEEE conference on computer vision and pattern recognition workshops, pp. 9-16.

46. Kollias, D., Tzirakis, P., Nicolaou, M. A., Papaioannou, A., Zhao, G., Schuller, B., and Zafeiriou, S. (2019). Deep affect prediction in-the-wild: Aff-wild database and challenge, deep architectures, and beyond. *International Journal of Computer Vision*, *127*(6-7), 907-929.

47. Pashine, S., Dixit, R., and Kushwah, R. (2021). Handwritten digit recognition using machine and deep learning algorithms. *arXiv preprint arXiv*, 2106.12614.

48. Nwankpa, C., Ijomah, W., Gachagan, A., and Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv*,1811.03378.

49. Udofia, U. (2018). *Basic overview of convolutional neural network (cnn)*. Retrieved May, 27, 2019.

50. Wayman, J., Jain, A., Maltoni, D., and Maio, D. (2005). *An introduction to biometric authentication systems*. In Biometric systems: Technology, design and performance evaluation (pp. 1-20). London: Springer London.

51. Minaee, S., Abdolrashidi, A., Su, H., Bennamoun, M., and Zhang, D. (2023). Biometrics recognition using deep learning: A survey. *Artificial Intelligence Review*, 1-49.

52. Yang, S., Luo, P., Loy, C. C., and Tang, X. (2016). *Wider face: A face detection benchmark*. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5525-5533.

53. Hjelmås, E., and Low, B. K. (2001). Face detection: A survey. *Computer vision and image understanding*, *83*(3), 236-274.

54. Zhang, C., and Zhang, Z. (2010). *A survey of recent advances in face detection*. Microsoft Corporation. One Microsoft Way Redmond, WA 98052 MSR-TR-2010-66.

55. Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. (2017, February). *Inception-v4, inception-resnet and the impact of residual connections on learning*. In Proceedings of the AAAI conference on artificial intelligence. Vol. 31, No. 1.

56. Ioffe, S., and Szegedy, C. (2015, June). *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. In International conference on machine learning, pp. 448-456.

57. O'Shea, K., and Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv*,1511.08458.

58. De Jong, K. (1988). Learning with genetic algorithms: An overview. *Machine learning*, *3*, 121-138.

59. Sun, Y., Xue, B., Zhang, M., Yen, G. G., and Lv, J. (2020). *Automatically designing CNN architectures using the genetic algorithm for image classification*. IEEE transactions on cybernetics, *50*(9), 3840-3854.

60. Internet: Tanmay Thaker. *Vgg16 easiest explanation*, (Aug 2021), Url: https://medium.com/nerd-for-tech/vgg-16-easiest-explanation-12453b599526. Last accessed date 03/11/2023.

61. Ramzan, F., Khan, M. U. G., Rehmat, A., Iqbal, S., Saba, T., Rehman, A., and Mehmood, Z. (2020). A deep learning approach for automated diagnosis and multi-class classification of Alzheimer's disease stages using resting-state fMRI and residual neural networks. *Journal of medical systems*, *44*, 1-16.

62. Cantor, S. B., and Kattan, M. W. (2000). Determining the area under the ROC curve for a binary diagnostic test. *Medical Decision Making*, *20*(4), 468-470.

63. Lasloum, T., Alhichri, H., Bazi, Y., and Alajlan, N. (2021). SSDAN: Multi-source semi-supervised domain adaptation network for remote sensing scene classification. *Remote Sensing*, *13*(19), 3861.

64. Ahmed, T., and Sabab, N. H. N. (2022). Classification and understanding of cloud structures via satellite images with EfficientUNet. *SN Computer Science*, *3*, 1-11.

65. Hu, J., Shen, L., and Sun, G. (2018). *Squeeze-and-excitation networks*. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7132-7141.

66. Yalman, Y., Uyanık, T., Atlı, İ., Tan, A., Bayındır, K. Ç., Karal, Ö. and Guerrero, J. M. (2022). Prediction of Voltage Sag Relative Location with Data-Driven Algorithms in Distribution Grid. *Energies*, *15*(18), 6641.

67. Kingma, D. P., and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv*,1412.6980.



Gazili olmak ayrıcalıktır