



Görüntü Üzerinden Uçak Tipini Belirleme

Yazılım Mühendisliği Ana Bilim Dalı

Yüksek Lisans Bitirme Projesi

Hüseyin Ersem Saygı

Y210240090

Danışman: Dr. Öğr. Üyesi Serpil Yılmaz

Haziran 2023

İzmir Kâtip Çelebi Üniversitesi Fen Bilimleri Enstitüsü öğrencisi **Hüseyin Ersem Saygı** tarafından hazırlanan Görüntü üzerinden nesne tespiti başlıklı bu çalışma tarafımızca okunmuş olup, kapsam ve nitelik açısından başarılı bulunarak tarafımdan YÜKSEK LİSANS PROJESİ olarak kabul edilmiştir.

ONAYLAYANLAR:

Danışman: Dr. Öğr. Üyesi Serpil Yılmaz
İzmir Kâtip Çelebi Üniversitesi

Yazarlık Beyanı

Ben, **Hüseyin Ersem Saygı**, Görüntü üzerinden nesne tespiti olan bitirme projemin ve projenin içinde sunulan bilgilerin şahsıma ait olduğunu beyan ederim. Ayrıca:

- Bu çalışmanın bütünü veya esası bu üniversitede Yüksek Lisans derecesi elde etmek üzere çalıştığım süre içinde gerçekleştirilmiştir.
- Daha önce bu bitirme projesinin herhangi bir kısmı başka bir derece veya yeterlik almak üzere bu üniversiteye veya başka bir kuruma sunulduysa bu açık biçimde ifade edilmiştir.
- Başkalarının yayımlanmış çalışmalarına başvurduğum durumlarda bu çalışmalara açık biçimde atıfta bulundum.
- Başkalarının çalışmalarından alıntıladığımda kaynağı her zaman belirttim. Yüksek lisans bitirme projesinin bu alıntılar dışında kalan kısmı tümüyle benim kendi çalışmamdır.
- Kayda değer yardım aldığım bütün kaynaklara teşekkür ettim.
- Bitirme projesinde başkalarıyla birlikte gerçekleştirilen çalışmalar varsa onların katkısını ve kendi yaptıklarımı tam olarak açıkladım.

Tarih: 06.06.2023

ÖZ

Bitirme projesi çalışması, uçak tipini belirlemek için yapay zeka temelli bir yöntemin geliştirilmesini amaçlamaktadır. Günümüzde havacılık sektöründe uçak tanıma süreçleri genellikle manuel olarak gerçekleştirilmekte ve uzmanlık gerektirmektedir. Bu süreçlerin otomatikleştirilmesi, hız, doğruluk ve verimlilik açısından önemli avantajlar sağlayabilir.

Projede, derin öğrenme ve görüntü işleme tekniklerinden faydalınalarak uçak görüntülerinden otomatik olarak uçak tipini belirleyebilen bir yapay zeka modeli geliştirilmiştir. Veri seti olarak geniş çaplı uçak görüntülerinden oluşan bir veri seti kullanılarak modelin eğitimi gerçekleştirilmiştir.

Geliştirilen model, uçak görüntülerini analiz ederek uçak tipini doğru bir şekilde sınıflandırabilme yeteneğine sahiptir. Modelin performansı, doğruluk oranı ve hata analizi yapılarak değerlendirilmiştir. Elde edilen sonuçlar, geliştirilen modelin yüksek doğruluk oranlarına ulaştığını ve uçak tipi belirleme konusunda etkili bir araç olduğunu göstermektedir.

Proje, havacılık sektöründe uçak tanıma süreçlerinin otomatikleştirilmesi ve iyileştirilmesi için bir adım oluşturmaktadır. Geliştirilen yapay zeka modeli, hava trafiği yönetimi, güvenlik kontrolü, uçak bakımı ve diğer havacılık süreçlerinde kullanılarak zaman ve maliyet tasarrufu sağlayabilir. Ayrıca, operasyonel verimliliği artırarak güvenlik standartlarının sağlanmasına da katkıda bulunabilir.

Anahtar Sözcükler: Uçak Tanıma, Havacılık Endüstrisi, Görüntü İşleme, Yapay Zeka, Derin Öğrenme

Abstract

This project aims to develop an artificial intelligence-based method for identifying aircraft types. In today's aviation industry, aircraft recognition processes are typically carried out manually and require expertise. Automating these processes can provide significant advantages in terms of speed, accuracy, and efficiency.

In this study, a deep learning-based artificial intelligence model is developed by leveraging deep learning and image processing techniques to automatically identify aircraft types from aircraft images. A comprehensive dataset of aircraft images is used for training the model.

The developed model has the ability to analyze aircraft images and accurately classify the aircraft type. The performance of the model is evaluated based on accuracy rates and error analysis. The results obtained demonstrate that the developed model achieves high accuracy rates and is an effective tool for aircraft type identification.

This study represents a step towards automating and improving aircraft recognition processes in the aviation industry. The developed artificial intelligence model can be used in air traffic management, security checks, aircraft maintenance, and other aviation processes, leading to time and cost savings. Additionally, it can contribute to enhancing operational efficiency and ensuring compliance with safety standards.

İçindekiler

Yazarlık Beyanı	ii
Öz	iii
Abstract	iv
1 Giriş	6
2 Nesne Tespitine Hazırlık.....	8
2.1 Veri Seti	8
2.2 Dataların Düzeltilmesi	8
2.3 Metrikler	9
2.3.1 İşProblemi.....	9
2.3.2 Uygun Metrik.....	10
2.4 CNN	11
2.4.1 Convolutional NeuralNetwork.....	11
2.4.1.1 İlkModelleme.....	12
2.4.1.2 Model.....	13
3 SONUÇLAR.....	15
3.1 Son Model.....	21
3.1.1 Son Model Sonuçları.....	26
Kaynaklar	32

Bölüm 1

Giriş

Yapay zeka ve görüntü işleme teknolojilerindeki hızlı gelişmeler, endüstrilerin birçok alanında çığır açıcı uygulamaların ortaya çıkmasına olanak sağlamaktadır. Bu teknolojilerin havacılık sektöründe kullanımı, uçak tespiti ve tanıma gibi alanlarda büyük önem taşımaktadır. Uçakların hızlı ve doğru bir şekilde tanınması, hava trafiği yönetimi, güvenlik kontrolü ve uçak bakımı gibi kritik süreçlerde büyük bir avantaj sağlamaktadır.

Bitirme projesi, uçak görüntülerinden yapay zeka kullanarak uçak tipini belirlemeyi amaçlamaktadır. Uçak tipinin otomatik olarak tanınması, manuel olarak yapılan tanıma süreçlerindeki hataları ve maliyeti azaltabilirken, daha hızlı ve güvenilir sonuçlar elde etmemizi sağlayacaktır. Ayrıca, uçak tipinin tanınması, havacılık endüstrisinde verimliliği artırabilir ve güvenlik standartlarının sağlanmasına yardımcı olabilir.

Projede, derin öğrenme tekniklerinin kullanıldığı bir yapay zeka modeli geliştirilmiştir. Model, geniş bir uçak veri seti üzerinde eğitilmiş ve uçak tipini belirleme yeteneği kazanmıştır. Veri seti, farklı uçak tiplerinden elde edilen gerçek dünya görüntülerinden oluşmaktadır. Görüntüler, önceden işleme ve veri artırma tekniklerine tabi tutulmuş ve modelin daha iyi performans elde etmesi sağlanmıştır.

Bu tez çalışmasının hedefleri şunlardır:

1. Uçak tipini belirlemek için bir yapay zeka modeli geliştirmek.
2. Modelin performansını değerlendirmek ve doğruluk oranını analiz etmek.
3. Modelin gerçek dünya uygulamalarında kullanılabilirliğini incelemek.

Çalışma, derin öğrenme, görüntü işleme ve yapay zeka alanlarındaki mevcut literatür ve teknikler üzerine yapılan derin bir literatür taramasıyla desteklenmektedir. Ayrıca, veri setinin toplanması, veri ön işleme süreçleri, model eğitimi ve sonuçların analizi gibi adımlar ayrıntılı olarak açıklanmıştır.

Proje, uçak tanıma teknolojilerinin geliştirilmesi ve havacılık sektöründeki uygulamalarının iyileştirilmesi için bir katkı sağlamayı hedeflemektedir. Uçak tipini belirleme konusundaki başarı, hava trafiği yönetimi, güvenlik kontrolü, uçak bakımı ve havacılık endüstrisindeki diğer süreçlerde önemli avantajlar sağlayabilir.

Bitirme projesi, bazı zorluklarla da karşılaşabilir. Uçakların farklı açılardan çekilmiş görüntülerinin olması, aydınlatma koşullarının değişkenliği ve farklı uçak modellerinin benzer görüntü özelliklerine sahip olması gibi faktörler, doğru uçak tipini belirlemeyi zorlaştırabilir. Bu nedenle, veri setinin çeşitliliği ve modelin genelleme yeteneği büyük önem taşımaktadır.

Bölüm 2

Nesne Tespitine Hazırlık

2.1 Veri Seti

Wu ve arkadaşları (2020) tarafından oluşturulan Çok Tipli Uçak Uzaktan Algılama Görüntüleri (MTARSI) veri seti kullanılacaktır. Bu veri seti 20 uçak tipine ait 9.385 görüntü içermektedir. CV kullanarak uçak tanıma için bir ölçüt olarak tasarlanmıştır. Görüntülerin bir kısmı Google görsellerinden elde edilmiş olsa da, çoğunluğu sentetiktir ve uçaklar farklı arka planlar üzerine bindirilerek ve üzerlerinde dönüşümler gerçekleştirilerek oluşturulmuştur.

Gelecekteki savaşlarda daha iyi mücadele edebilmek için veri toplamak ve daha iyi modeller oluşturmak amacıyla daha fazla çatışma aramak mantığa aykırı olacaktır, çünkü bir ordunun ilk ve en önemli amacı çatışmaları en başta önleyen caydırıcılıktır. Burada önemli olan nokta, sentetik veri ve simülasyonun YZ yeteneklerinin oluşturulması ve test edilmesi için kilit yetenekler olduğudur.

2.2 Dataların Düzeltilmesi

Setin içerisinde bazı isimlendirilme hataları bulunmaktadır. Bunlardan bazıları F-15 F16 şeklinde, F-4 F-22 şeklinde isimlendirildiği görülmektedir. Doğru etiketler atandıktan sonra, veri kümesinde 21 sınıfa dağıtılmış 9.320 görüntü kalmıştır. Çeşitli görüntülerden bir örnek şekil 1 de verilmiştir:



Şekil 1.Uçak Tipleri

2.3 Metrikler

2.3.1 İş Problemi

Hava taşıtlarının görüntülerini üretmek için halihazırda bir sistem olduğunu varsayıyoruz. Bir operatörün daha sonra bir uçak tipini ve konumunu etiketlemek için bunları elemesi gerekir, fakat burada konumu göz ardı edilmektedir. Bu nedenle CV modeli, uçak tipi için önerilerde bulunarak zaman ve bilişsel kapasite tasarrufu sağlayarak değer sunar.

Modelin hata yapacağından emin olduğumuz için, döngüdeki kişinin bu hataların ne zaman meydana gelebileceğinin farkında olmasını sağlamalıyız. Bu, modelin her bir tahmindeki güvenini ,örneğin operatör tarafından tanımlanan eşiklere göre yüksek, orta veya düşük,görüntüleyerek başarılabilir. Bu şekilde, operatör (kullanıcı) [1] modelin tipik olarak iyi performans gösterdiği sınıflar ve [2] belirli bir güven seviyesinin üzerindeki tahminler için öneriyi otomatik olarak kabul edebilirken, belirli bir güven seviyesinin altındaki tahminlerde bir çağrı yapmak için devreye girebilmektedir.

Operatör ayrıca üst düzey avcı uçakları (örn. F-22) gibi belirli uçak tipleri için de tetiktedir. Bu nedenle, model (a) bir F-22'yi düşük bir güven seviyesiyle tahmin ettiğinde veya (b) tipik olarak F-22'lerle karıştırdığı uçakları (örneğin F-16) tahmin ettiğinde bir uyarı bayrağı üretmelidir.

2.3.2 Uygun Metrik

Modelin görüntülerdeki uçaklar için iyi tavsiyelerde bulunma becerisine odaklanılmaktadır. Bu nedenle, ilk 1 doğruluk, ilk 2 doğruluk ve ilk 3 doğruluk değerlerini kullanılacaktır. Bunlar, gerçek etiketin tahmin edilen ilk n sınıf içinde olup olmadığını ölçer [3].

Basitlik açısından, bir ISR ekibinin günde 1.500 görüntü işlediği varsayılmaktadır, bu da saatte yaklaşık 60 veya dakikada 1 görüntü anlamına gelmektedir [4].

Denetimli otonom etiketleme (man-on-the-loop), sistem görüntüleri otomatik olarak en iyi tahmin edilen sınıfla etiketler ve etiket yanlış olduğunda operatör devreye girer. Burada Top-1 doğruluğu önemlidir.

Yarı-otonom etiketleme , tahmin edilen olasılık %99'dan büyükse sistem görüntüleri otomatik olarak en üst tahmin edilen sınıfla etiketler. Aksi takdirde, ilk 3 sınıfı sunar ve operatörün doğru etiketi girmesi gerekir. İlk 3 sınıfın doğru etiketi içermesi durumunda operatörün neredeyse hiç çaba harcamayacağı varsayılmaktadır. Her zaman, etiket yanlışsa veya öneriler doğru uçak tipini içermiyorsa operatör devreye girmelidir. Bu nedenle, burada ilk 1 doğruluk ve ilk 3 doğruluğun bir kombinasyonu söz konusudur.

2.4 CNN

Convolutional Neural Network (Evrifimli sinir ađları) yapay sinir ađı kullanılacaktır.

2.4.1 Convolutional Neural Network

Convolutional Neural Network (CNN), özellikle grnt tanıma ve iřleme gibi alanlarda kullanılan bir yapay sinir ađı trdr. CNN'ler, veri ierisindeki desenleri algılamak ve zelliklerini ıkarmak iin zel olarak tasarlanmıřlardır.

Bir CNN, genellikle  temel bileřenden oluřur: evriřim (convolution), aktivasyon fonksiyonları ve pooling.

- Evriřim (Convolution): Evriřim iřlemi, grnt veya veri zerinde filtrelerin kaydırılarak uygulanmasıdır. Bu filtreler, verinin belirli zelliklerini vurgulamak veya ıkarmak iin kullanılır. rneđin, kenarları algılamak iin bir kenar algılama filtresi kullanılabilir. Evriřim katmanları, grnt zerindeki lokal bađlantıları keřfetmek ve nemli zellikleri ne ıkarmak iin etkili bir yntemdir.
- Aktivasyon Fonksiyonları: Her evriřim katmanından sonra, bir aktivasyon fonksiyonu kullanılır. Aktivasyon fonksiyonları, sinir ađına non-lineerlik ekler ve ıktıları sıkıřtırır. Bu, sinir ađına daha fazla esneklik sađlar ve daha karmařık desenleri renmesini sađlar. Sigmoid, ReLU (Rectified Linear Unit) ve tanh gibi yaygın aktivasyon fonksiyonları kullanılabilir.
- Pooling: Pooling, veri boyutunu kltmek ve ıktıların boyutunu azaltmak iin kullanılan bir iřlemdir. En yaygın kullanılan pooling tr, en byk deđeri veya ortalama deđeri olarak veriyi zetler. Bu, ađın leklenebilirliđini artırır ve hesaplama ykn azaltır.

CNN'ler, genellikle evriřim katmanları, aktivasyon fonksiyonları ve pooling katmanlarından oluřan tekrarlayan bir yapıda tasarlanır. Bu yapı, verinin daha karmařık zelliklerini katmanlar arasında renmesini ve sonunda sınıflandırma veya tanıma gibi bir ıktı retmesini sađlar [7].

CNN'ler, görüntü sınıflandırması, nesne tespiti, yüz tanıma, otomatik sürüş gibi birçok görevde başarılı sonuçlar vermiştir. Bu nedenle, derin öğrenme alanında önemli bir yer tutarlar.

2.4.1.1 İlk Modelleme

Genel olarak, test edilen her model için özel olarak ayarlanan öğrenme oranı dışında hiçbir şey değiştirilmemiştir.

- Girdi boyutu: 70x70
- Tüm evrişimli katmanlar için:
 1. Çekirdek boyutu 3x3
 2. ReLu aktivasyonu

- Tüm evrişimli bloklar için:
 1. 2x2 havuz boyutu ile Max pooling
 2. Her maksimum havuzlama katmanından sonra toplu normalleştirme
- Tam bağlantılı katman(lar) için:
 1. 0,5 düğümün bırakılması
 2. ReLu aktivasyonu

- Modelin derlenmesi:
 1. Kayıp: Kategorik çapraz entropi (standart bir kayıp metriği)
 2. İyileştirici: Adam 0.0001 öğrenme oranı ile (bazı deneylerden sonra bulundu)
 3. Uyum:
 4. 16'lık parti boyutu (bazı deneylerden sonra bulundu)
 5. Sabırlı 50 epok ile erken durdurma (bazı deneylerden sonra bulundu)

2.4.1.2 Model

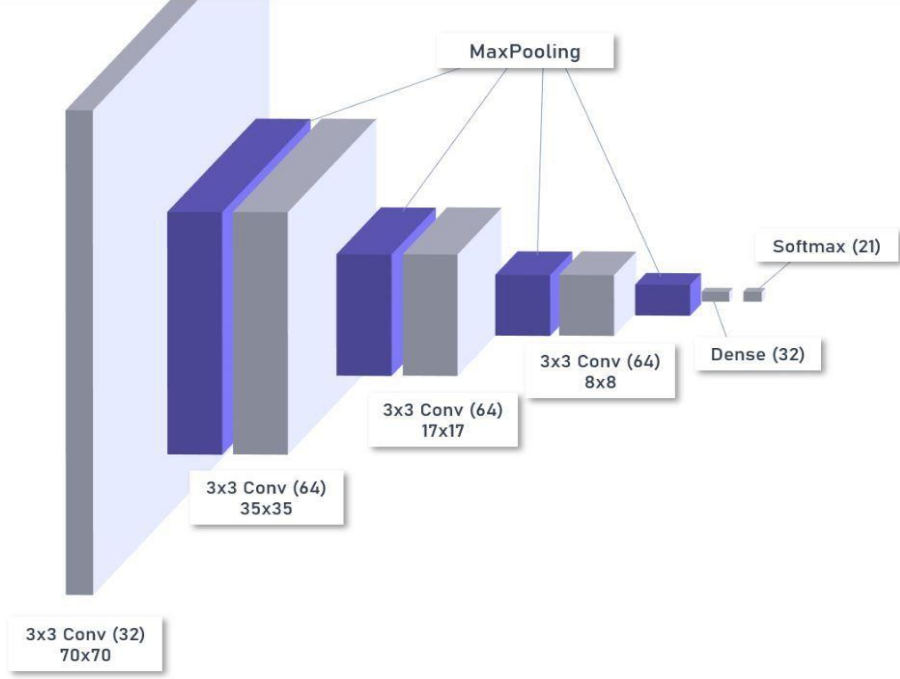
```
3. # Modeli tanıt
4. model = Sequential()
5. # Blok 1
6. model.add(Conv2D(filters=32, kernel_size=(3,3), activation='relu', padding='same', kernel_regularizer=L2(l2=0.01), input_shape=(70, 70, 3)))
7. model.add(MaxPool2D(pool_size=(2,2)))
8. model.add(BatchNormalization())
9.
10. # Blok 2
11. model.add(Conv2D(filters=64, kernel_size=(3,3), activation='relu', padding='same', kernel_regularizer=L2(l2=0.01)))
12. model.add(MaxPool2D(pool_size=(2,2)))
13. model.add(BatchNormalization())
14.
15. # Blok 3
16. model.add(Conv2D(filters=64, kernel_size=(3,3), activation='relu', padding='same', kernel_regularizer=L2(l2=0.01)))
17. model.add(MaxPool2D(pool_size=(2,2)))
18. model.add(BatchNormalization())
19.
20. # Blok 4
21. model.add(Conv2D(filters=64, kernel_size=(3,3), activation='relu', padding='same', kernel_regularizer=L2(l2=0.0001)))
22. model.add(MaxPool2D(pool_size=(2,2)))
23. model.add(BatchNormalization())
24.
25.     # Dense
26.     model.add(Flatten())
27.     model.add(Dense(32, activation='relu'))
28.     model.add(Dropout(0.5))
29.     model.add(Dropout(0.5))
30.     # Softmax
```

```

31.     model.add(Dense(21, activation='softmax'))
32.     # Derleme
33.     model.compile(
34.         loss='categorical_crossentropy',
35.         optimizer=Adam(learning_rate=0.0001),
36.         metrics=['accuracy']
37.     )

```

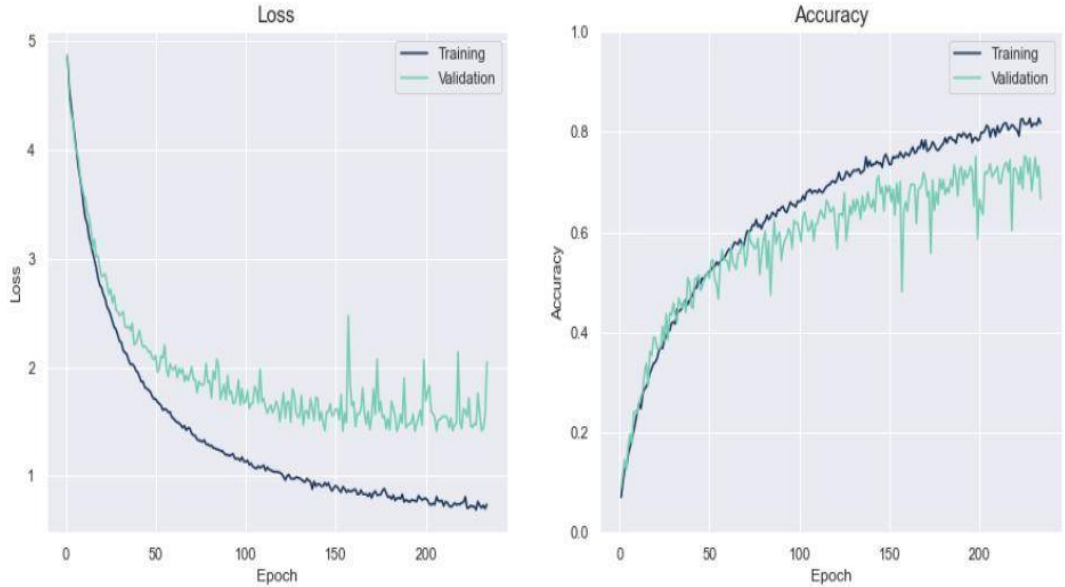
Layer (type)	Output Shape	Param #
conv2d_8 (Conv2D)	(None, 70, 70, 32)	896
max_pooling2d_8 (MaxPooling2D)	(None, 35, 35, 32)	0
batch_normalization_8 (Batch Normalization)	(None, 35, 35, 32)	128
conv2d_9 (Conv2D)	(None, 35, 35, 64)	18496
max_pooling2d_9 (MaxPooling2D)	(None, 17, 17, 64)	0
batch_normalization_9 (Batch Normalization)	(None, 17, 17, 64)	256
conv2d_10 (Conv2D)	(None, 17, 17, 64)	36928
max_pooling2d_10 (MaxPooling2D)	(None, 8, 8, 64)	0
batch_normalization_10 (Batch Normalization)	(None, 8, 8, 64)	256
conv2d_11 (Conv2D)	(None, 8, 8, 64)	36928
max_pooling2d_11 (MaxPooling2D)	(None, 4, 4, 64)	0
batch_normalization_11 (Batch Normalization)	(None, 4, 4, 64)	256
flatten_2 (Flatten)	(None, 1024)	0
dense_4 (Dense)	(None, 32)	32800
dense_5 (Dense)	(None, 21)	693



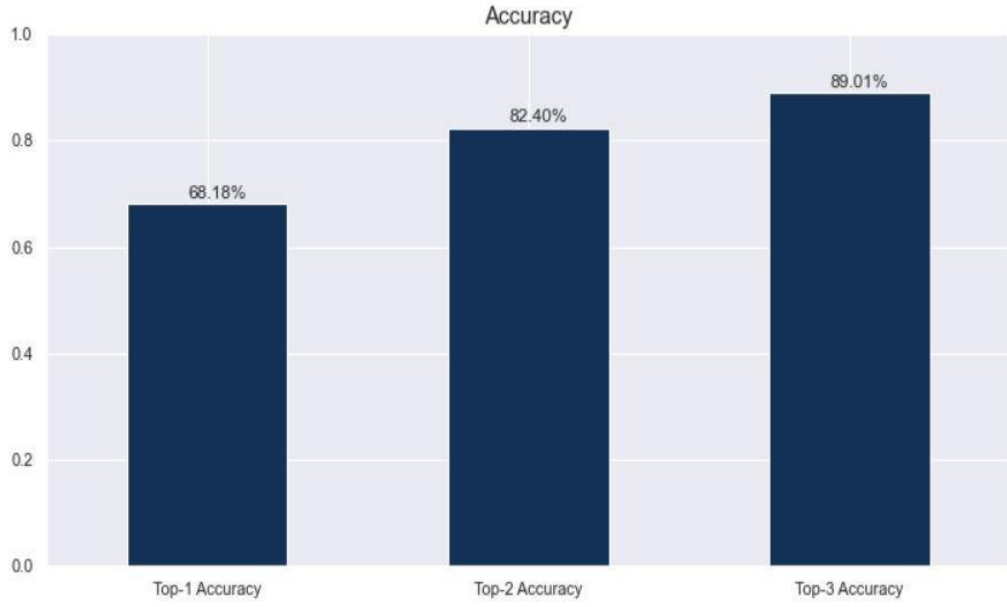
Şekil2.Max Pooling

3 SONUÇLAR

Şekil 4’te görüldüğü gibi ilk olarak, doğruluk oranının %68 olduğunu, ilk 2 ve ilk 3 doğruluk oranlarının ise sırasıyla %82 ve %89 görülmektedir.



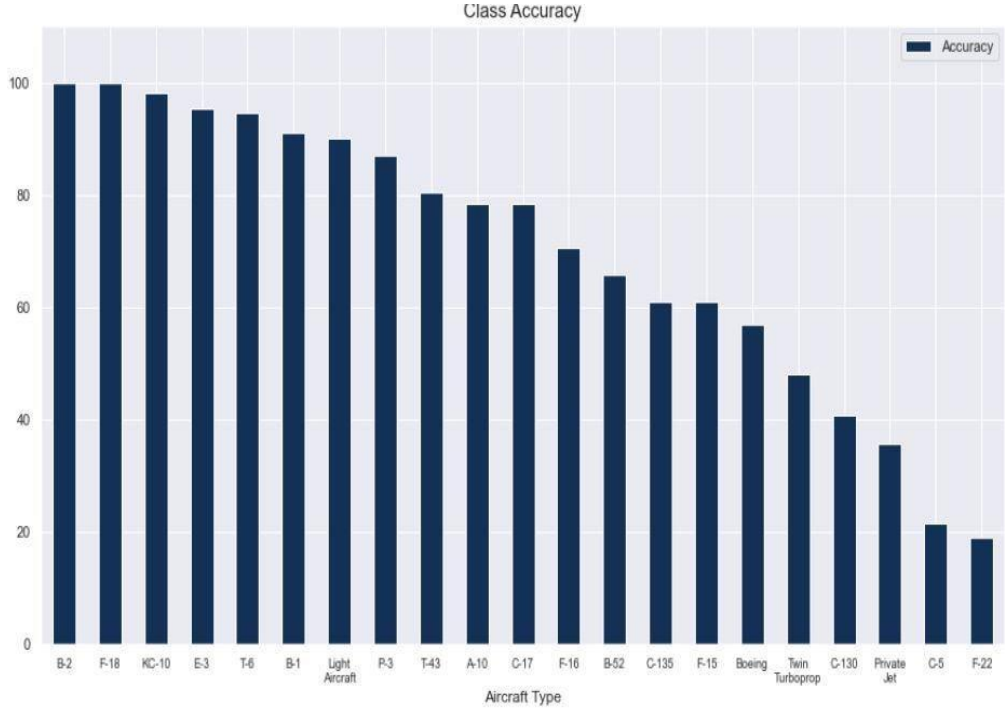
Şekil3.Sonuçların Doğruluk Oranları



Şekil 4.Doğruluk Oranları

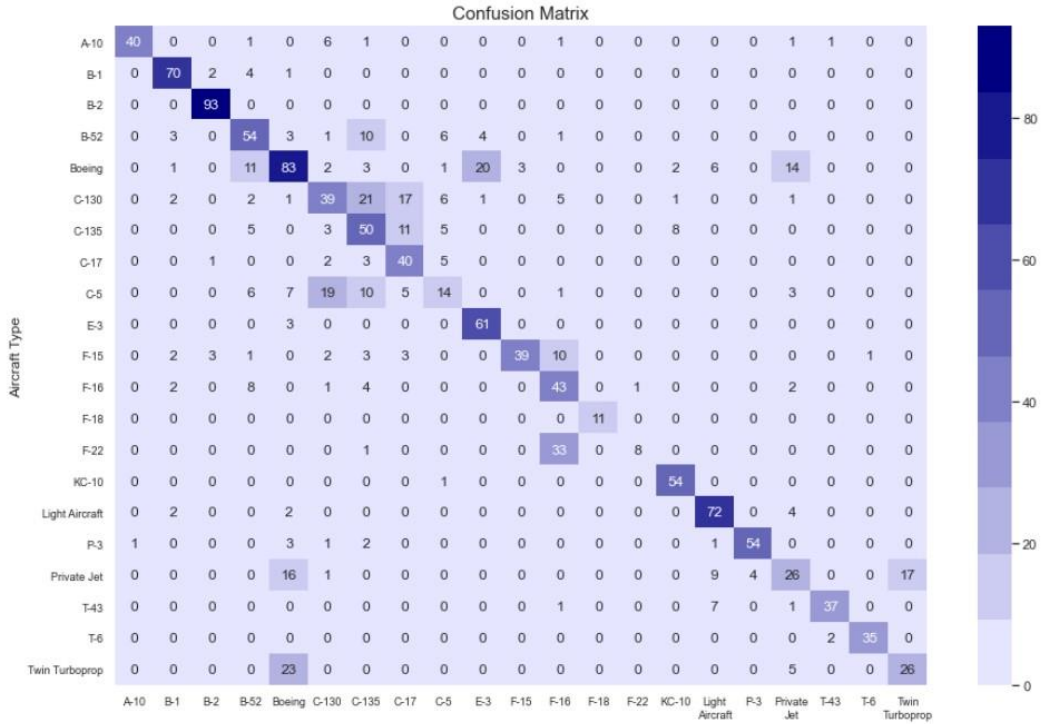
- 1.Denetimli otonom: Sistem %68 doğruluk oranına ulaşacak ve operatör saatte 60 görüntünün %32'sini veya 19'unu manuel olarak etiketlemek zorunda kalacaktır.
- 2.İnsan-makine ekibi %89'luk bir doğruluk oranına ulaşacaktır:
- 3.Doğru olan otomatik etiketler: %27
- 4.Önerilen doğru etiketler: %62,
- 5.Operatörün saatte 60 görüntünün %11'ini veya yaklaşık 6,6'sını manuel olarak etiketlemesi gerekecektir.

İkinci olarak, modelin iyi performans gösterdiği uçak tiplerine bakıldığında ,sonuçlar oldukça sezgisel görünmektedir. B-2, E-3 ve T-6'nın hepsi çok benzersiz özelliklere sahiptir. F-18'lerin ilk birkaç sırada yer almasının nedeni muhtemelen görüntülerdeki çeşitlilik eksikliğidir. Hafif uçaklar hantal, kare şekilleri nedeniyle oldukça iyi bir şekilde öne çıkmaktadır.



Şekil 5.Uçaklara göre doğruluk değerleri

Üçüncü olarak, bazı sınıflardaki düşük performansı anlamak için karışıklık matrisini incelememiz gerekmektedir.Şekil-6 da bu durum gösterilmektedir.



Şekil 6.Karışıklık Matrisi

Karışıklıklardan bazıları şunlardır:

- Boeing'ler (sivil uçaklar) B-52'ler, E-3'ler ve Özel Jetler ile karıştırıldı
- C-130'lar C-135'ler ve C-17'lerle karıştırıldı
- C-5'ler C-130'lar ve C-135'lerle karıştırıldı
- F-15'ler F-16'larla karıştırıldı, ama tersi olmadı
- Garip bir şekilde, F-16'lar B-52'lerle karıştırıldı.
- F-22'ler F-16'larla fena halde karıştırıldı, ama tersi olmadı
- Özel Jetler İkiz Turboprop'lar, Hafif Uçaklar ve Boeing'lerle karıştırıldı
- İkiz Turboprop'lar Boeing'lerle karıştırıldı

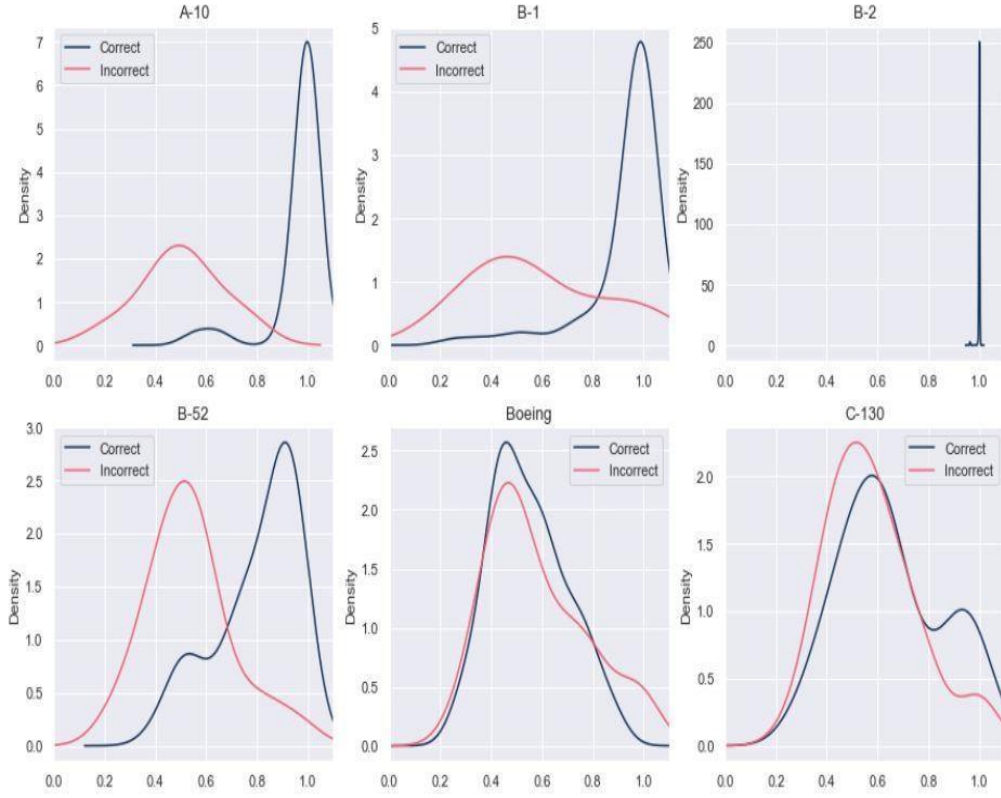
Mantıken, bir ISR operatörü modelin gerçekte olduğundan daha az tehlikeli bir uçak türünü tahmin etmesini istemeyecektir. Operatörler savaş uçaklarını daha hızlı ve ölümcül oldukları için daha çabuk tespit etmek istemektedirler. Buna karşılık, bir nakliye uçağının tespit edilmesinde gecikme olursa başları belaya girmeyebilir çünkü bunlar daha yavaş ve tespit edilmesi daha kolaydır. Eğer operatör gerçekte bir F-16 varken modelin B-52 tahminini kabul ederse, meşru bir tehdide gecikmeli bir yanıt verilebilmektedir.

Dördüncü olarak, modelin tahminlerde bulunurken ne kadar emin olduğuna dair bir değerlendirme gerekmektedir. İdeal olarak, model doğru olduğunda yüksek olasılıklar, yanlış olduğunda ise düşük olasılıklar üretilmesi gerekmektedir. Bu değerlendirmeyi yapmak için iki tür grafik kullanırız [5] :

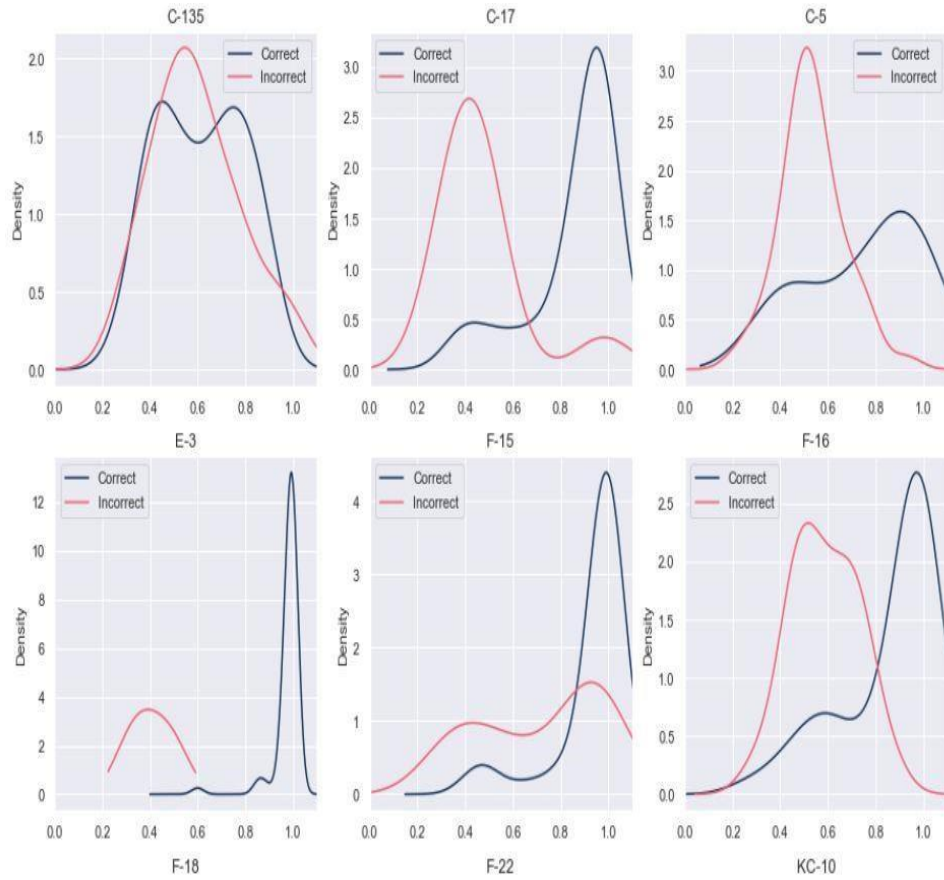
1.Doğru ve yanlış tahminler için ortalama tahmin edilen olasılığın çubuk grafiği. Bu bize uçak tipleri arasındaki güvene genel bir bakış sağlamaktadır.

2.Doğru ve yanlış tahminler için tahmin edilen olasılıkların yoğunluk grafikleri. Bu bize dağılımlardaki kesin farkı gösterir.

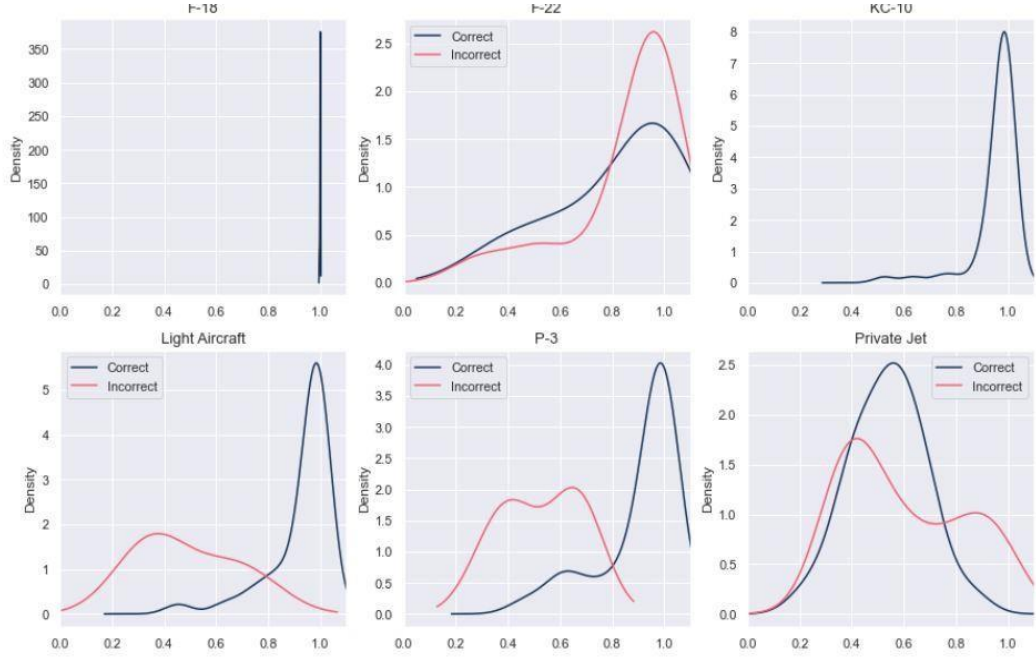
Genel olarak, modelin nispeten kötü performans gösterdiği uçak tipleri için tahmin edilen olasılıkların dağılımlarında daha düşük farklılaşma görülmektedir: Boeing, C130, C-135, C-5, F-15, F-22, Özel Jet ve İkiz Turboprop. (Bknz. Şekil 7-8)



Şekil 7.Dağılım Grafikleri



Şekil 8.Dağılım Grafikleri



Şekil 9. Dağılım Grafikleri

Genel olarak, model kesinlikle denetimli otonom modda yeterince iyi performans göstermemektedir. Bununla birlikte, yarı otonom modda çalıştırılırsa ISR operatörüne biraz zaman ve çaba kazandırabilir.

Modelin muhtemelen uçak tanıma konusunda tamamen eğitimsiz birinden daha iyi performans göstermektedir. Mantıken, eğitimsiz bir kişi uçakları kategorize etmek için basit bir sezgisel yöntem kullanır.

- Nakliye uçağı: Çok fazla şey taşıyabilecekmiş gibi görünen geniş kanat açıklığına sahip büyük uçaklar
- Hafif uçak: Nispeten küçük ve zararsız/hantal görünümlü.
- Savaş uçağı: Nispeten küçük, keskin ve vahşi görünümlü.
- Diğer: Sadece A-10 yukarıdaki kategorilerden herhangi birine sığdırılmaz. Hafif uçak ve nakliye uçağının bir melezesidir.

Her kategoriden en yaygın sınıfı tahmin ettiğini varsayıldığı düşünüldüğünde (nakliye uçağı için C-130, hafif uçak olarak kare şeklinde herhangi bir şey, avcı uçakları için

F-15 ve diğerleri için A-10), %25-30'luk bir doğruluk elde edilecektir. Buna karşılık, model %68'lik bir doğruluk oranına ulaşmaktadır.

3.1 SON MODEL

Son model, 13 ağırlık katmanında 12 milyon parametre ile yapılmaktadır.

```
# Initialise model
model = Sequential()

# Block 1
model.add(Conv2D(filters=64, kernel_size=(3,3), activation='relu', padding='same', kernel_regularizer=L2(l2=0.01), input_shape=(70, 70, 3)))
model.add(Conv2D(filters=64, kernel_size=(3,3), activation='relu', padding='same', kernel_regularizer=L2(l2=0.01)))
model.add(Conv2D(filters=64, kernel_size=(3,3), activation='relu', padding='same', kernel_regularizer=L2(l2=0.01)))
model.add(MaxPool2D(pool_size=(2,2))) model.add(BatchNormalization())

# Block 2
model.add(Conv2D(filters=128, kernel_size=(3,3), activation='relu', padding='same', kernel_regularizer=L2(l2=0.01)))
model.add(Conv2D(filters=128, kernel_size=(3,3), activation='relu', padding='same', kernel_regularizer=L2(l2=0.01)))
model.add(Conv2D(filters=128, kernel_size=(3,3), activation='relu', padding='same', kernel_regularizer=L2(l2=0.01)))
model.add(MaxPool2D(pool_size=(2,2))) model.add(BatchNormalization())
```

```

# Block 3

model.add(Conv2D(filters=256, kernel_size=(3,3), activation='relu', padding='same', kernel_regularizer=L2(l2=0.01)))

model.add(Conv2D(filters=256, kernel_size=(3,3), activation='relu', padding='same', kernel_regularizer=L2(l2=0.01)))

model.add(Conv2D(filters=256, kernel_size=(3,3), activation='relu', padding='same', kernel_regularizer=L2(l2=0.01)))

model.add(MaxPool2D(pool_size=(2,2))) model.add(BatchNormalization())

# Block 4

model.add(Conv2D(filters=512, kernel_size=(3,3), activation='relu', padding='same', kernel_regularizer=L2(l2=0.01)))

model.add(Conv2D(filters=512, kernel_size=(3,3), activation='relu', padding='same', kernel_regularizer=L2(l2=0.01)))

model.add(Conv2D(filters=512, kernel_size=(3,3), activation='relu', padding='same', kernel_regularizer=L2(l2=0.01)))

model.add(MaxPool2D(pool_size=(2,2))) model.add(BatchNormalization())

# Dense model.add(Flatten())

model.add(Dense(512, activation='relu'))

model.add(Dropout(0.5))

# Softmax model.add(Dense(21,

activation='softmax'))

# Compile

```

```

model.compile(
loss='categorical_crossentropy',
optimizer=Adam(learning_rate=0.00004),
metrics=['accuracy']
)

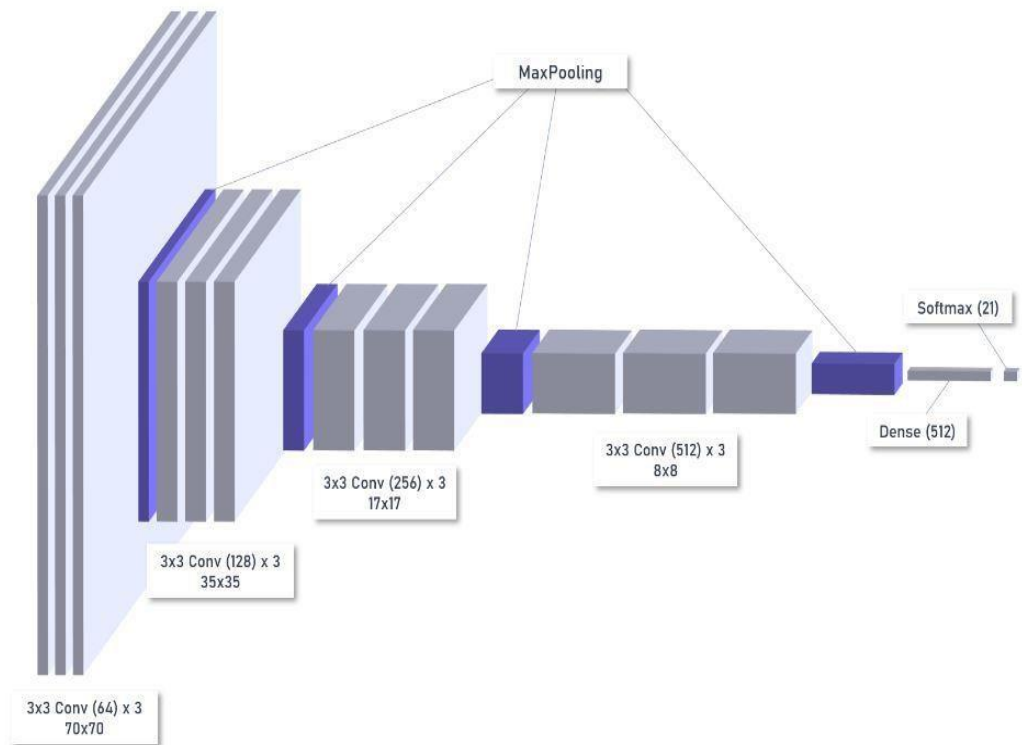
```

Layer (type)	Output Shape	Param #	
(Conv2D)	(None, 70, 70, 64)	1792	conv2d
conv2d_1 (Conv2D)	(None, 70, 70, 64)	36928	
conv2d_2 (Conv2D)	(None, 70, 70, 64)	36928	
max_pooling2d (MaxPooling2D)	(None, 35, 35, 64)	0	
batch_normalization (BatchNo	(None, 35, 35, 64)	256	
conv2d_3 (Conv2D)	(None, 35, 35, 128)	73856	
conv2d_4 (Conv2D)	(None, 35, 35, 128)	147584	
conv2d_5 (Conv2D)	(None, 35, 35, 128)	147584	
max_pooling2d_1 (MaxPooling2	(None, 17, 17, 128)	0	
batch_normalization_1 (Batch	(None, 17, 17, 128)	512	

conv2d_6 (Conv2D)	(None, 17, 17, 256)	295168
<hr/>		
conv2d_7 (Conv2D)	(None, 17, 17, 256)	590080
<hr/>		
conv2d_8 (Conv2D)	(None, 17, 17, 256)	590080
<hr/>		
max_pooling2d_2 (MaxPooling2D)	(None, 8, 8, 256)	0
<hr/>		
batch_normalization_2 (Batch Normalization)	(None, 8, 8, 256)	1024
<hr/>		
conv2d_9 (Conv2D)	(None, 8, 8, 512)	1180160
<hr/>		
conv2d_10 (Conv2D)	(None, 8, 8, 512)	2359808
<hr/>		
conv2d_11 (Conv2D)	(None, 8, 8, 512)	2359808
<hr/>		
max_pooling2d_3 (MaxPooling2D)	(None, 4, 4, 512)	0
<hr/>		
batch_normalization_3 (Batch Normalization)	(None, 4, 4, 512)	2048
<hr/>		
flatten (Flatten)	(None, 8192)	0
<hr/>		
dense (Dense)	(None, 512)	4194816
<hr/>		
dense_1 (Dense)	(None, 21)	10773
<hr/>		
=====		
Total params:		12,029,205

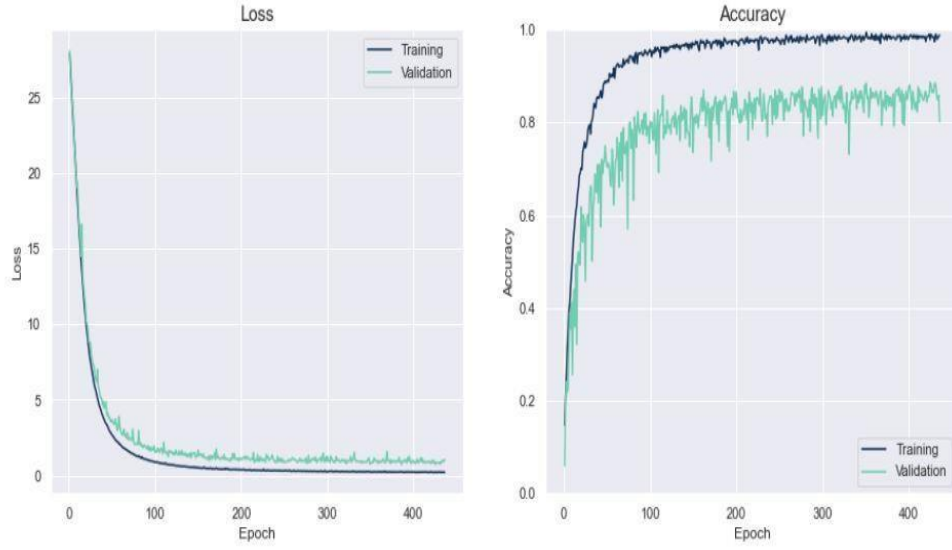
Trainable params: 12,027,285

Non-trainable params: 1,920

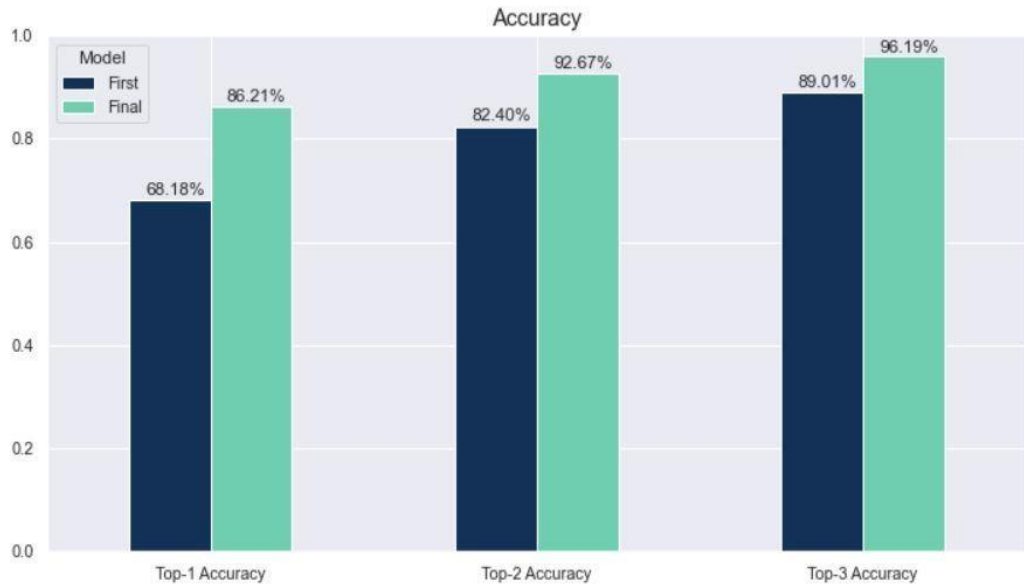


Şekil 10.Max Pooling

2.1.1 SON MODELİN SONUÇLARI



Şekil 11.Doğruluk Oranları



Şekil 12.Doğruluk Oranları

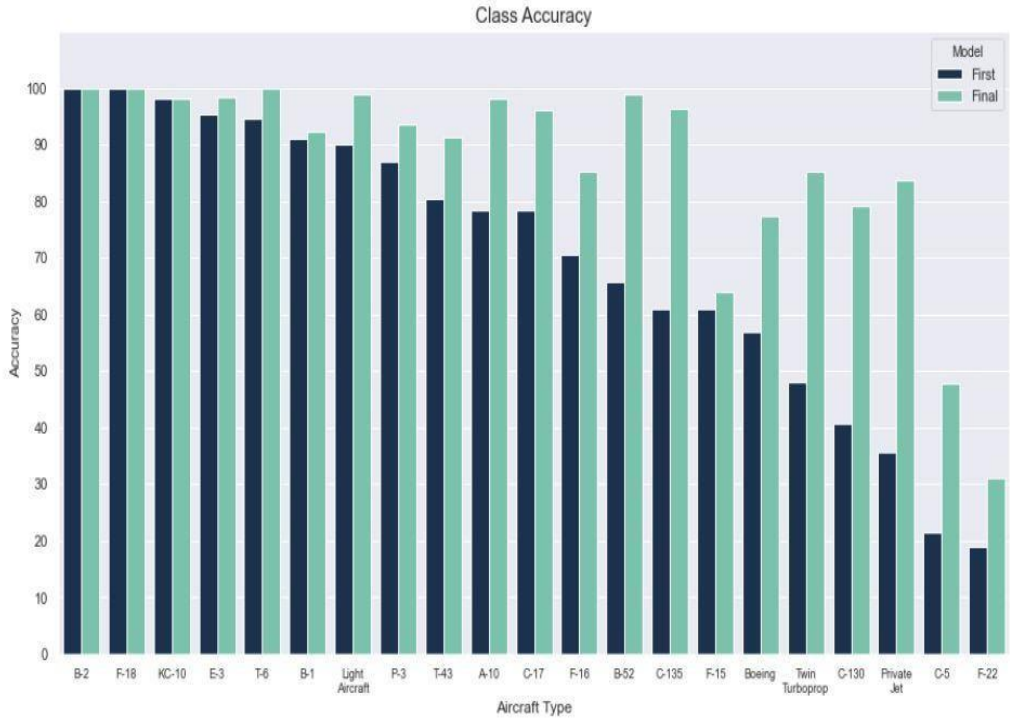
Model denetimli otonom ve yarı otonom modda çalışıyor ise:

- 1.Denetimli otonom: Sistem %86 doğruluk oranına ulaşacak ve operatörün saatte 60 görüntünün %14'ünü veya yaklaşık 8,4'ünü manuel olarak etiketlemesi gerekecektir
- 2.İnsan-makine ekibi %96'lık bir doğruluk oranına ulaşacaktır: Doğru olan otomatik etiketler: %66

3.Önerilen doğru etiketler: %30

4.Operatörün saatte 60 görüntünün %4'ünü veya 2,4'ünü manuel olarak etiketlemesi gerekecektir.

İkinci olarak, özellikle daha önce iyi performans göstermeyen uçak tiplerinde sınıf doğruluğunun genel olarak iyileştiği görülmektedir. Hala düşük doğruluğa sahip olan F-15, F-22 ve C-5 için daha fazla çalışma yapılabilir ve daha fazla veri eklenebilir. (Bknz.Şekil 13)

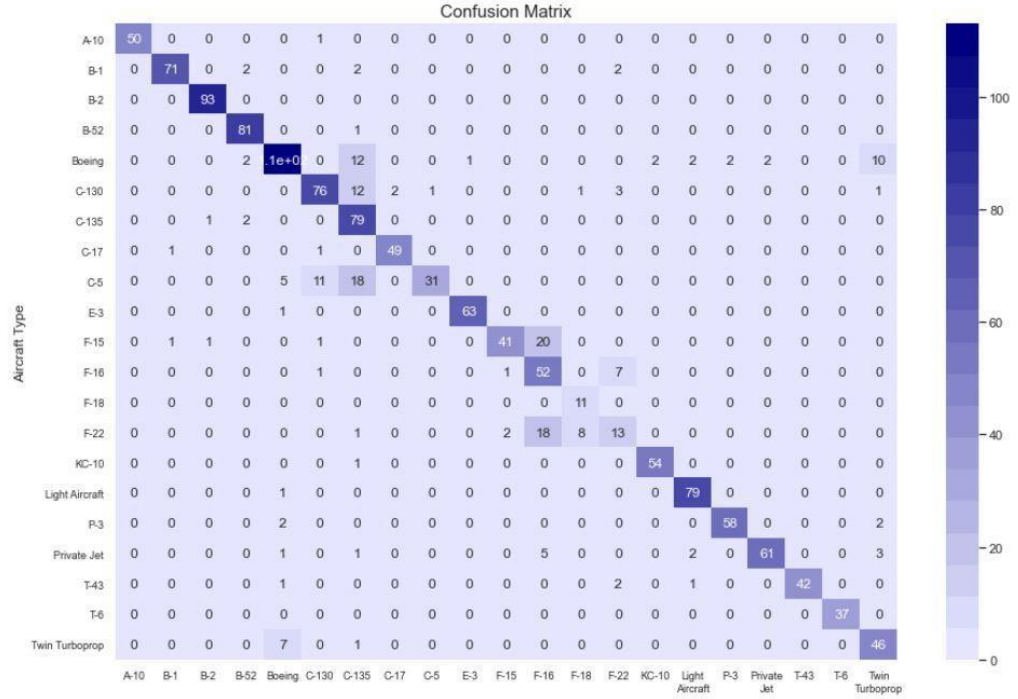


Şekil 13.Uçakların Doğruluk Oranları

Üçüncü olarak, "karışıklıklardaki" gelişmeleri kontrol etmek için karışıklık matrisini incelenecektir. Genel olarak, avcı uçakları birbirleriyle karıştırılmakta ve nakliye uçakları için de aynı şey geçerlidir. Artık avcı uçakları ile nakliye uçaklarını karıştırma sorunu bulunmamaktadır.(Bknz. Şekil-14)

1. Boeing'ler (sivil uçaklar) artık C-135'ler ve Twin Turboprop'lar ile karıştırılmaktadır
2. C-130'lar artık sadece C-135'lerle karıştırılmaktadır

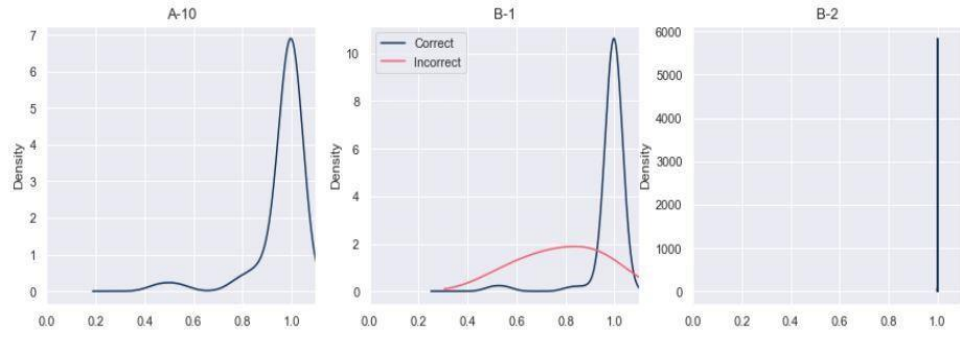
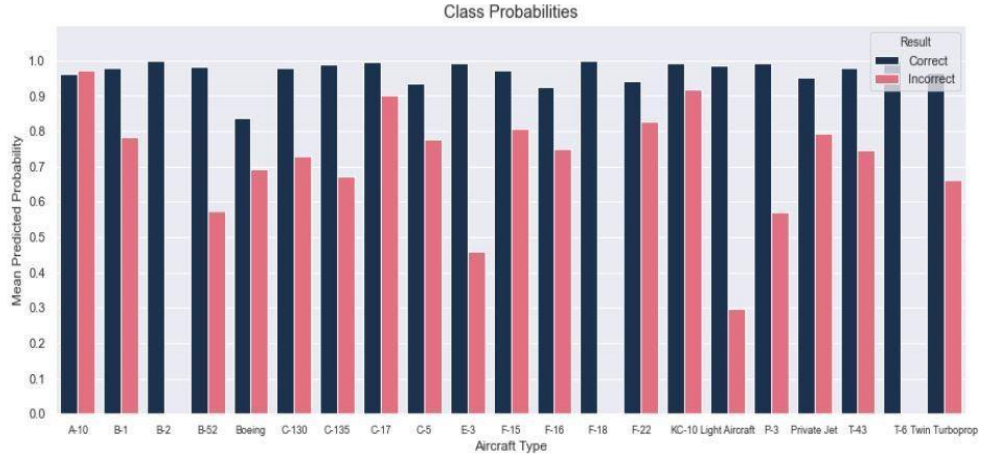
3. C-5'ler hala C-130'lar ve C-135'lerle karıştırılmakta
4. F-15'ler hala F-16'larla karıştırılmakta,
5. F-16'lar artık B-52'lerle karıştırılmakta
6. F-22'ler hala F-16'larla ve şimdi de F-18'lerle karıştırılmakta
7. İkiz Turboprop'lar hala Boeing'lerle karıştırılmaktadır.



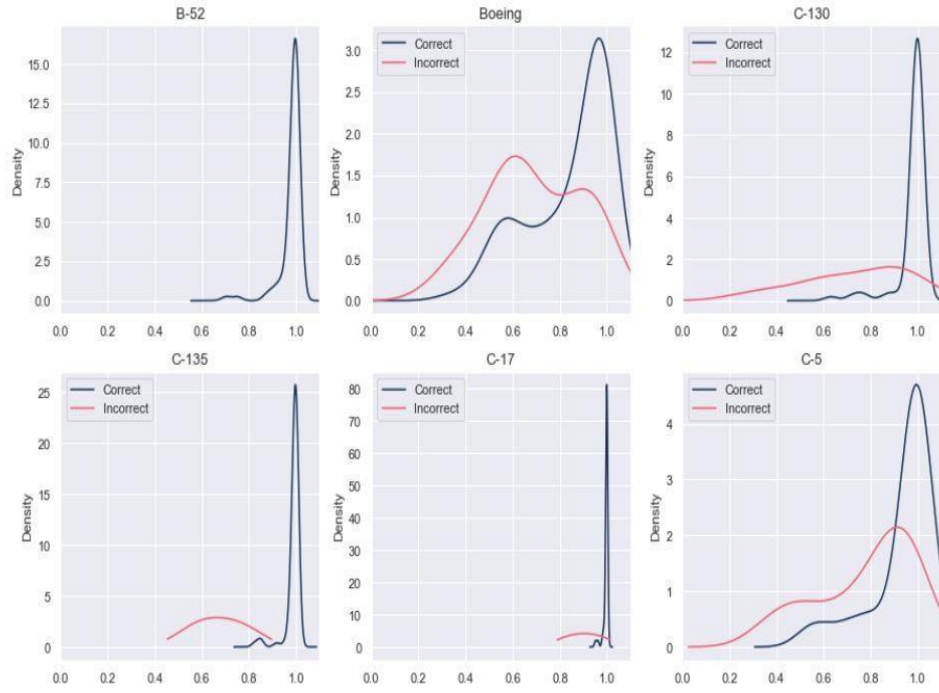
Şekil 14. Karışıklık Matrisi

Modelin nispeten kötü performans gösterdiği uçak tipleri için tahmin edilen olasılıkların dağılımlarındaki farklılaşmada bazı iyileşmeler görülmektedir. Boeing, C-130, C-135, C-5, F-15, F-22, Özel Jet ve İkiz Turboprop. F-22 üzerinde en çok çalışılması gereken uçak tipi olmaya devam etmektedir.

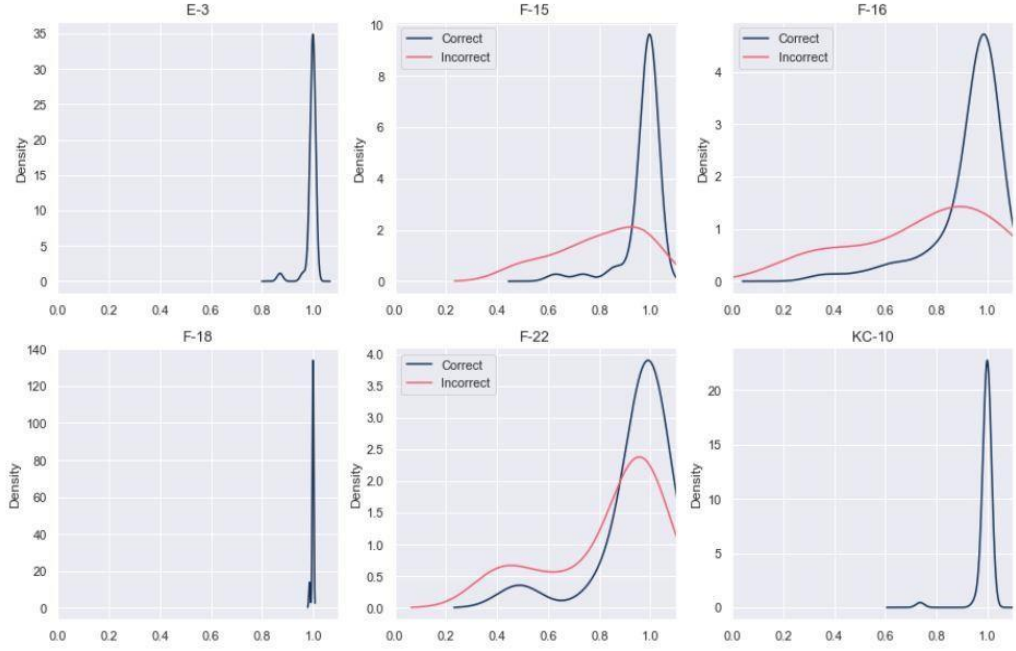
Ayrı olarak, çubuk grafikte (ilk grafik) A-10 için doğru ve yanlış tahmin edilen olasılıkların ortalama değerinin neredeyse aynı olduğuna dikkat çekmektedir. Bunun nedeni, C-130'un A-10 olduğuna dair tam olarak tek bir yanlış tahminde bulunulmuş olmasıdır. (Bknz. Şekil -15, Şekil -16, Şekil -17, Şekil -18)



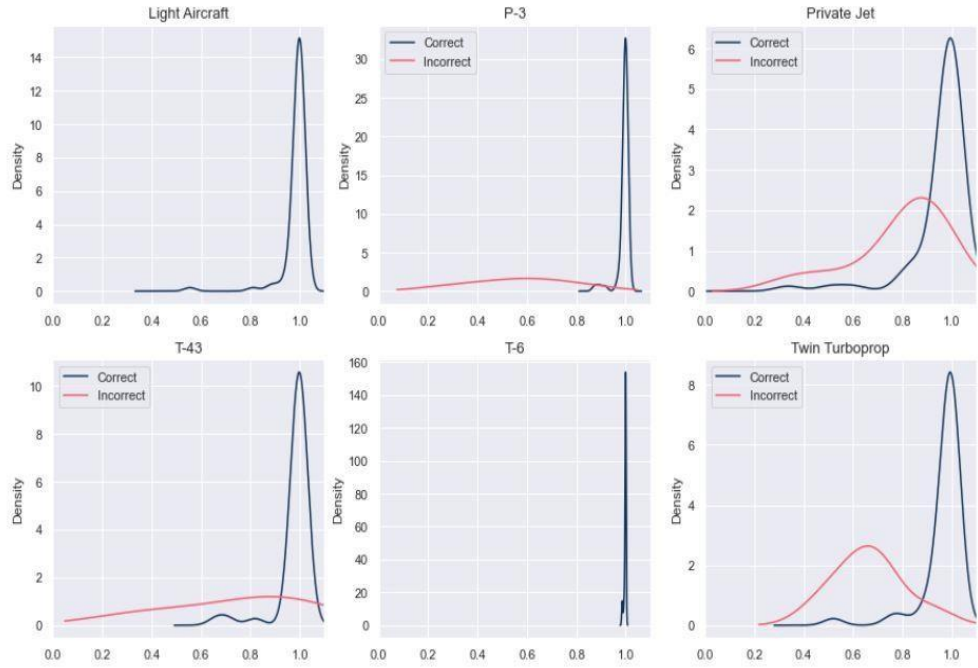
Şekil 15.Olasılık Doğruluk Grafikleri



Şekil 16.Olasılık Doğruluk Grafikleri



Şekil 17.Olasılık Doğruluk Grafikleri



Şekil 18.Olasılık Doğruluk Grafikleri

hava aractanımı için bilgisayarla görmeyi kullanma iş problemi anlatılmıştır, ilk ve son modeller gösterilmiştir ve bir sinir ağına ine ayar yapmak için bazı öğrenme ipuçları paylaşılmıştır. Bu modelin hava aractanımı konusunda ISR operatörlerine ne kadar yardımcı olabileceğine [6] ilişkin asıl soruna dönülürse, bir operatörün gerçekleştirmesi gerekebilecek manuel işlem miktarına ilişkin kabaa bir tahmin kullanılmıştır. Eğer nihai model denetimli otonom ve yarı otonom modda çalıştırılırsa:

Denetimli otonom etiketleme (man-on-the-loop):

1. Sistem görüntülerin %86'sını otomatik olarak doğru şekilde etiketleyecektir
2. Operatörün görüntülerin kalan %14'ünü manuel olarak etiketlemesi gerekecektir; bu da saatte 60 görüntüden 8,4'üne denk gelmektedir
3. Genel olarak, insan-makine ekibi görüntülerin %96'sını doğru şekilde etiketlemiştir
4. 66'sı otomatik olarak doğru şekilde etiketlenecektir (tahmin edilen olasılık \geq %99)
5. 30 model tarafından doğru bir şekilde tavsiye edilecektir
6. Operatörün görüntülerin kalan %4'ünü manuel olarak etiketlemesi gerekecektir; bu da saatte 60 görüntüden 2,4'üne denk gelmektedir
7. Modeller bazı iyi sonuçlar üretmiş olsa da, modelin dahil edilip edilmemesi konusunda hala bir değerlendirme yapılması gerekmektedir. İşlerin nasıl yapıldığına dair sorular sormamız gerekiyor bu operatörlerin cevaplayabileceği sorulardır.

Örneğin:

- Ne kadar doğruluk gerekiyor?
- Sonuçlara ne kadar çabuk ihtiyacınız var?
- Hangi hedefler daha çok ilgi çekiyor?
- Bu hedefler için ne düzeyde güvene ihtiyacınız var?
- Operatör arayüzle nasıl etkileşime giriyor?

Bu temel hususlar veri bilimi uzmanlığının dışında kalsa da, dahil edilmeleri zor değildir.

Kaynaklar

- [1] **J.S. Chen, A. Iluertas and G. Medioni, "Fast Convolution with Laplacian-ofGaussian Masks", IEEE Pat. Analys. and Mach. Intell., Vol. 9, pp. 584-590, 1987.**
- [2] **Marr and E.C. Hildreth, "Theory of Edge Detection", Royal Soc. London, pp. 187- 217, 1980.**
- [3] **G.E. Sotak and K.L. Boyer, "The Laplacian-ofGaussian Kernel: A Formal Analysis and Design Procedure for Fast, Accurate Convolution and Full-Frame Output", Comp. Vision, Graphics and Image Processing, Vol. 48, pp. 147-189, 1989.**
- [4] **T. Lindeberg, "Detecting Salient Blob-Like Image Stuctures and Their Scales with a ScaleSpace Primal Sketch: A Method for Focus-ofAttention", Int. I. Computer Vision, Vol. 11, pp. 283-318, 1993.**
- [5] **T. Lindeberg, "Edge Detection and Ridge Detection within Automatic Scale Selection", IEEE Conf Computer Vision and Pattern Recognition, pp. 465-470, 1996.**
- [6] **A. Khashman and K.M. Curtis "Scale Space Analysis Applied to 3Dimensional Object Recognition", E-LETTER on Digital Signal Processing, Georgia Institute of Technology, Atlanta, USA, Issue No. 24, June 1995.**
- [7] **https://en.wikipedia.org/wiki/Convolutional_neural_network**