# Face and Face-mask Detection System using VGG-16 Architecture based on Convolutional Neural Network

Chamandeep Vimal
Research Scholar,
Department of Computer Science and Engineering,
IES, IPS Academy, Indore, 452012, India

Neeraj Shirivastava
Associate Professor,
Department of Computer Science and Engineering,
IES, IPS Academy, Indore, 452012, India

## ABSTRACT
Face recognition can be used in several applications such as in surveillance, identification in login system and personalized technology. The challenge of the face detection system is the non-frontal face position and the use of accessories that cover the face area; even conventional detection systems that rely on facial features are difficult to get high accuracy. The proposed system can overcome these problems and it can detect human face with mask also. The deep learning system can recognize facial features with complex backgrounds. The VGG16 architecture based on convolutional neural network with shallow layers to produce light computing then the system can work real-time. Multiple layer detection on the last feature map is used to detect varied face sizes. The system result shows sequential images of face localization with 93% accuracy.

## Keywords
Face detection, Real-time, CPU, Multiple layer, Deep learning

## 1. INTRODUCTION
Face detection is an artificial intelligence based computer technology used to find and identify human faces in digital images and it's also called facial detection. Face detection technology can be used in various fields such as biometrics, security, law enforcement, entertainment, and personal safety to provide real-time surveillance and tracking of people. Corona virus illness (COVID-19) is the most recent outbreak caused by the recently identified corona virus. COVID-19 is an infectious illness caused by the SARS-CoV-2 virus that affects the respiratory system. It is mostly transmitted from person to person by airborne transmission, particularly through close contact. Because of the COVID-19 epidemic, the WHO has published numerous preventative measures to combat the spread of corona virus. The most visible principles are social distance, sanitization, and masking. Wearing a face mask inhibits the corona's spread across the population. As a result, the majority of countries have implemented mandatory face mask legislation in public places. Manually inspecting the face mask is a time-consuming process, especially in crowded areas like hospitals, airports, train stations, and retail malls. This prompted researchers to develop an automated face mask detecting system. Inception Net transfer learning was used by Jignesh et al. [1] to recognize face masks. Loey et al. [2] presented a hybrid deep transfer learning model for detecting face masks. In a similar vein, we presented VGG-16 architecture model for face mask recognition.

## 2. LITRATURE SERVEY
In previous works, facial detection and recognition is based on the standard two-step machine learning approach, in which a variety of special characteristics or attributes are derived from the images in the first step, and in the second step, a classifier is used to classify the emotions [3-5]. The most recent couple of years addressed a thriving time of examination in the space of human face detection and recognition. This research exploration helped human in many applications such as security applications, automobiles, photographic applications etc. Human facial features extraction (like nose, eyes, mouth) and preparing of the models on those highlights are the initial steps of facial recognition and detection. Viola-Jones illustrated the traditional conventional approach [6]. This approach has resulted in a robust and fast face detection system at some points, but its accuracy is low, it was a major weakness of this approach. It can improvise its accuracy of detection and classification significantly with the use of convolutional neural network (CNN). Several researchers have used it to determine the location of the face in an image [7-9]. Convolutional neural network (CNN) is the best approach to find out facial features accurately and it has a ability to distinguish the face from the background features displayed in map features [10]. Heavy computing on CNN seems like a challenge when the system applied in real-time, several methods and techniques have found solutions to this problem [11-12]. However, working rapidly is also insufficient as these technologies are incorporated in low-cost embedded systems such as the CPU. We need a CNN machine that can operate in real-time on the CPU while maintaining acceptable accuracy.

## 3. SYSTEM MODEL
Using the Python programming language, CNN model is created. Utilized Tensor-Flow, an open-source software framework commonly used for machine learning applications such as neural networks, and Keras, which acts as a wrapper and is a high-level neural network framework built on top of Tensor-Flow.

## 4. PROBLEM STATEMENT
In this paper, work on an image classification issue with a limited amount of training examples per category. The data repository contains images of human faces with mask and without mask, and the goal of the assignment is to create a heuristic and robust model that can detect human face with mask and without mask.

## 5. PROPOSED SOLUTION
Firstly, use a Convolutional neural network, in which the input pictures are processed through a succession of layers, including convolutional, pooling, flattening, and fully connected layers, and then the output of CNN is formed, which classifies pictures. After building CNN models from the ground up, use image augmentation techniques to fine-tune the model. As a result, use one of the pre-trained model VGG-16 to categorise images and assess accuracy for training and validation data.

## 5.1 Convolutional Neural Network

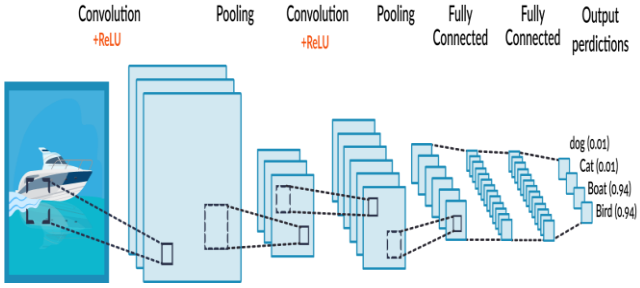


**Figure 1: Architecture of Convolutional Neural Network**

A Convolutional neural network is a sort of artificial neural network that employs multiple perceptrons to evaluate picture inputs and have learnable weights and bases to several sections of pictures that may separate each other. The usage of Convolutional Neural Networks has the benefit of leveraging the usage of local spatial coherence in the input pictures, which allows them to have fewer weights because certain parameters are shared. In terms of memory and complexity, this procedure is definitely efficient. The following are the main building components of a convolutional neural network:

### 5.1.1 Convolution layer

A kernel matrix is passed over the input matrix in the convolutional layer to build a feature map for the following layer. By sliding the Kernel matrix over the input matrix, we perform a mathematical action known as convolution. At each point, an element-wise matrix multiplication is done and the resulting total is shown on the feature map. Convolution is a type of linear operation that is frequently utilized in a range of disciplines such as image processing, statistics, and physics. Convolution can be used on more than one axis. If we have a 2-Dimensional image input, I, and a 2-Dimensional kernel filter, K, we may compute the convoluted picture as follows:

$$S(i,j) = \Sigma\Sigma I(m,n)k(i-m,j-n)$$

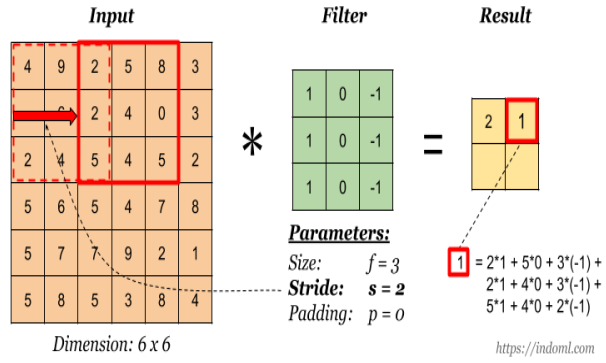


**Figure 2: Element wise matrix multiplication and summation of the results onto feature map in convolutional layer.**

### 5.1.2 Non-linear Activation Function

The activation function is the nonlinear modification that we do on the input signal. It is a node that comes after the convolutional layer. The rectified linear unit activation function (ReLU) is a piecewise linear function that gives outputs the input if it is 1, if it's not 1, it outputs zero.

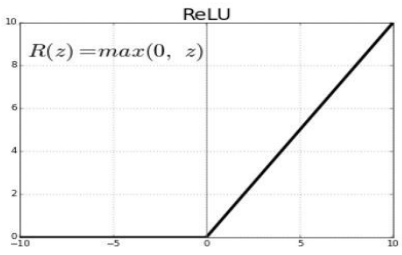


**Figure 3: ReLU activation function**

### 5.1.3 Pooling layer

The disadvantage of the convolutional layer's feature map output is that it stores the precise position of features in the input. This implies that any cropping, rotation, or other tiny adjustments to the input picture will result in an entirely new feature map. To address this issue, we propose down sampling of convolutional layers. Applying a pooling layer after the nonlinearity layer allows for down sampling. Pooling aids in making the representation roughly invariant to tiny translations of the input. Translation invariance states that if we translate the input by a little amount, the values of the majority of the pooled outputs do not change.

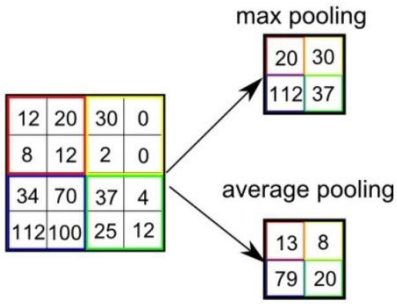


**Figure 4: Max pooling and Average pooling**

### 5.1.4 Fully-connected layer

The output of the Final Pooling Layer serves as input to the Fully Connected Layer at the conclusion of a convolutional neural network. One or more of these levels may exist. Every node in the first layer is connected to every node in the second layer if it is fully linked.

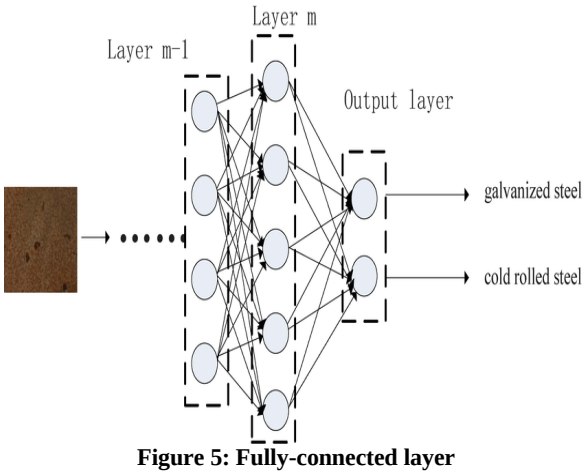


**Figure 5: Fully-connected layer**

## 5.2 VGG-16 Architecture

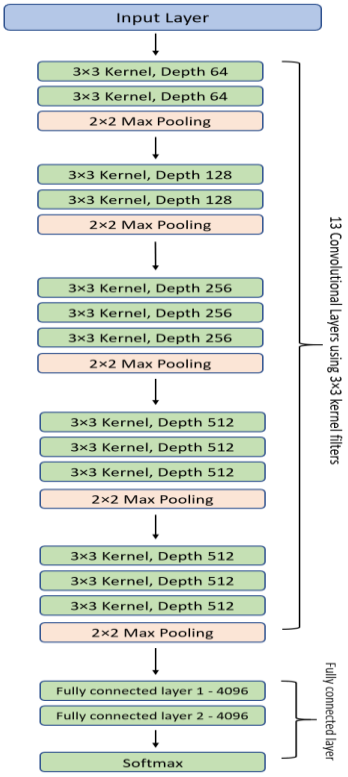


**Figure 6: VGG-16 Model Architecture**

There are 13 convolutional layers, two fully linked layers, and one Softmax classifier. Karen Simonyan and Andrew Zisserman published Very Deep Convolutional Network for Large Scale Image Recognition in 2014, which introduced the VGG-16 architecture. Karen and Andrew built a 16-layer network using convolutional and fully linked layers. For simplicity, just 3x3 convolutional layers were placed on top of each other.

The precise structure of the VGG-16 network shown in Fig. 6 is as follows:

The first and second convolutional layers are made up of 64 feature kernel filters with a filter size of 3x3. The dimensions of the input picture (RGB picture with depth 3) change to 224x224x64 as it passes through the first and second convolutional layers. The output is then sent to the max pooling layer with a stride of 2.

The third and fourth convolutional layers are made up of 124 feature kernel filters with a filter size of 3x3. Following these two layers is a max pooling layer with stride 2, and the resultant output is 56x56x128.

Convolutional layers with kernel size 3x3 are used in the fifth, sixth, and seventh levels. All three make use of 256 feature maps. Following these layers is a max pooling layer with stride 2.

Eighth to thirteenth are two groups of convolutional layers with kernel size 3x3. All of these convolutional layer sets contain 512 kernel filters. Following these layers is a max pooling layer with a stride of 1.

Fourteen and fifteen levels are completely linked hidden layers of 4096 units, followed by a softmax output layer (sixteenth layer) of 1000 units.

## 5.3 Proposed Model

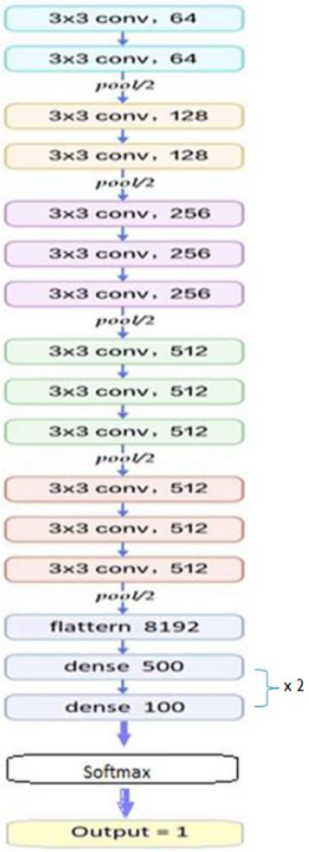


**Figure 7: Proposed Architecture Diagram**

The proposed model use VGG16 architecture and transfer learning, which is a process of applying the knowledge of weights and layers from a current model to new untrained model in order to accelerate the learning of a new model. VGG16 (also called Oxford-Net) is a convolutional neural network architecture named after the Visual Geometry Group from Oxford, who developed it. It was used to win the ILSVR (ImageNet) competition in 2014. In proposed model existing VGG16 is used for transfer learning based face detection and recognition and proposed model conducts train and validation on Face Detection custom data set with varying facial image position, lighting, and size. The images on the data set contain thousands faces, thousand of images with mask and without mask and also used weights of Imagenet that provide better results. VGG16 is a convolution based network model. VGG16 is a network model focused on convolutions and it is widely used in computer vision technologies. In proposed model for better classification, removed the fully connected layer top most layer of existing model and replaced this layer with the flatten, dense and dense softmax layer for better classification. To prevent over fitting, the drop out is used to drop certain values at random. For multi-classification of facial features in an image softmax layer is used. Except for the last layer, the ReLu activation mechanism has been used in all layers. The number of images used in the training phase 80% of the total number of images in the dataset, with the remainder used for confirmation. During the preprocessing step, all RGB images are resized 128 x 128. The following are the subjects of train and validation experiments: Google Colab is a virtual science technology simulation powered by a Genuine Intel CPU @ 2.20 GHz, 13 GB RAM, and a 16 GB Tesla V100 GPU.

## 6. DATASET

The training archive contains cumulatively 10000 images of human faces with a mask and without a mask. In this paper, our objective is to build a robust image classification model with restrictions consisting of few training samples of faces with mask and without a mask. We need to downscale the number of images to add a constraint on the input images. We took 5000 data images of faces with a mask and without a mask each. From the dataset, we took 80% images for training and 20% of images for validation. We leverage the pre-trained VGG-16 model as a feature extractor and transfer low-level features, such as edges, corners, rotation, and learn new level features specific to the target problem which is to classify the images by giving a matrix, we used python openCV for real-time face detection where if a face is not covered with the mask it can create a red box, but if the face is covered with the mask it creates a green box with mask label.

(a)

(b)

**Figure 8: (a) With mask data set images, (b) Without mask data set images**

## 7. ANALYSIS

As previously mentioned, we will begin by creating a basic convolutional neural network from scratch, then train it using a training picture dataset and assess the model. We will later enhance the accuracy by utilizing an image augmentation approach. Finally, we will extract features and categorise pictures using the pre-trained model VGG-16, which has previously been trained on a custom dataset with a broad variety of categories.

The key stage in this phase is the CNN preparation process. In the training process, the proposed model employs random weights as the initial weight. Learning demonstrates that the machine can understand the distinguishing features of the face. It learns the key symmetrical structure of a face in an input image. The input training batch size is 32 images with 25 number of epoch, multi-scale training plans are implemented. This system's effect will learn each input picture for a different scale we used image data generator for image augmentation which perform different scaling operation on whole dataset, fig.9 shows some examples of augmentation operation on image.

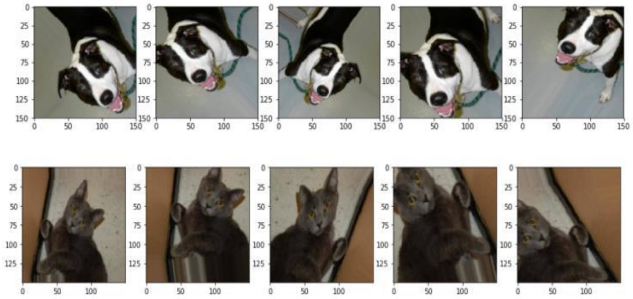

**Figure 9: Examples of an Images are populated with zoom, rotation, height, width after applying ImageDataGenerator ().**

The first and second convolutional layers consist 28x128x64 feature kernel filters followed by kernal size 3x3. As input image passed into first and second convolutional layer, dimensions change to 128x128x64. Then the resulting output is passed to max pooling layer with a stride of 2. The third and fourth convolutional layers consist of 64x64x128 feature kernel filters. These two layers are followed by a max pooling layer with stride 2. The fifth, sixth and seventh layer consist of 32x32x256 kernal filters followed by kernal size 3x3, and these layers followed by max pooling layer with stride 2. The eighth, ninth and tenth layer consist of 16x16x512 kernal filters followed by kernal size 3x3, and these layers followed by max pooling layer with stride 2. Then the next 4 layers consist of 8x8x512 kernal filters followed by kernel size 3x3. Follow by max-pooling with stride 1. The output of last layers is then passed to input to the Flatten layer followed by dense layers with scale (500, 100, 1). The model is trained on VGG16 architecture transfer learning technique by loading the weights of "Imagenet". The model will identify complex shape of a face with and without mask face with the best accuracy of one in the first detection layer. However, other layers are not shown this good accuracy of detection on medium and small faces ($2^{nd}$ detection layer=0.8, and $3^{rd}$ detection layer=0.75). A collection of validation images from the dataset is used to validate the model of the training outcomes. We use validation loss followed by training loss as an accuracy metrics to determine how precise our model accuracy is. This procedure is used to determine how precise the training results are. This score shows that how the proposed model can correctly predict the faces in an image.

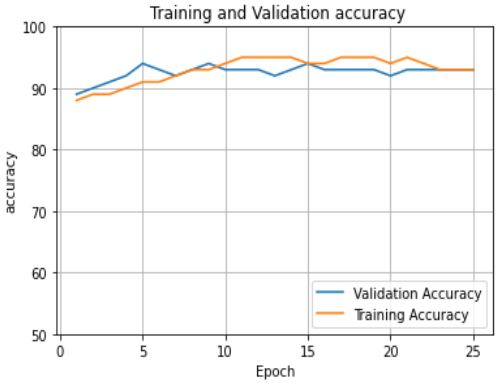


**Figure 10: Training and Validation Accuracy of proposed model**

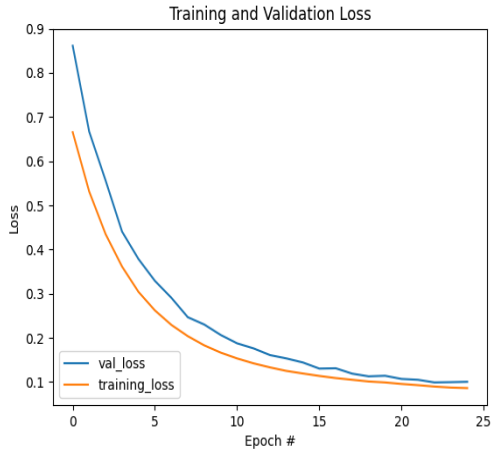


**Figure 11: Training and Validation losses of proposed model**

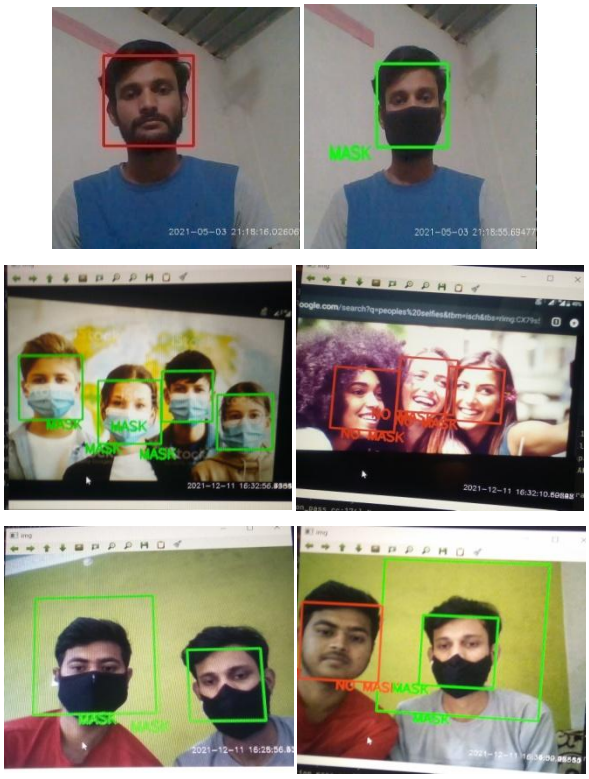

**Figure 12: Face and Face Mask Detecting Image**

**Table 1: Comparison Table of used models in the Project**

| **Models** | **Training Accuracy** | **Validation Accuracy** |
|---|---|---|
| Basic Convolutional Neural Network | 87% | 74% |
| Fine tuned CNN with image Augmentation | 85% | 80% |
| Proposed Model | 95% | 93% |

**Table 2: This table shows the precision, recall and F1score values of proposed model**

| **Model** | **Precision** | **Re-call** | **F1-Score** |
|---|---|---|---|
| Proposed Model | **94 %** | **95%** | **94%** |

# 8. CONCLUSION

Face recognition can be used in several applications such as in surveillance, identification in the login system, and personalized technology. The challenge of the face detection system is the non-frontal face position and the use of accessories that cover the face area. This paper presents the real-time face detection system that can overcome the difficulties of non-frontal face detection with better accuracy. This system detects multiple human faces at a time and also when the face is covered with a mask. The sensible outcomes are generated when the framework is tried on different postures of faces. This work illustrates the model's preliminary steps toward a high accuracy of the real-time face detection device. In future work, making the model more precise and gaining more accuracy and efficiency, model train on a larger dataset ( dataset of 25k images approx) and also work on face key points and a better camera will be used for capturing face images in real-time, so the system can work on deem light also.

# 9. REFERENCES

[1] G. J. Chowdary, N. S. Punn, S. K. Sonbhadra and S. Agarwal, Face mask detection using transfer learning of inceptionv3, 2020.

[2] M. Loey, G. Manogaran, M. H. N. Taha and N. E. M. Khalifa, "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the covid-19 pandemic", *Measurement*, vol. 167, pp. 108288, 2021.

[3] Yamashita, R., Nishio, M., Do, R.K.G. *et al.* Convolutional neural networks: an overview and application in radiology. *Insights Imaging* **9,** 611–629 (2018). https://doi.org/10.1007/s13244-018-0639-9.

[4] Z. Zhang, M.J. Lyons, M. Schuster, S. Akamatsu, Comparison be- tween geometry-based and Gabor-wavelets-based facial expression recognition using multi-layer perceptron, in: IEEE International Con- ference on Automatic Face & Gesture Recognition (FG), 1998.

[5] Y. Tian, Evaluation of face resolution for expression analysis, in: CVPR Workshop on Face Processing in Video, 2004.

[6] M. Abdulrahman and A. Eleyan, "Facial expression recognition us- ing Support Vector Machines," *2015 23nd Signal Processing and Com- munications Applications Conference (SIU)*, Malatya, 2015, pp. 276-279.

[7] Paul Viola , Michael Jones, "Robust Real-time Face Detection", International Journal of Computer Vision 57(2), 137–154, 2004.

[8] S. Saypadith and S. Aramvith, "Real-Time Multiple Face Recognition using Deep Learning on Embedded GPU System," 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Honolulu, HI, USA, 2018, pp. 1318-1324.

[9] Ya Wang, Tianlong Bao, Chunhui Ding and Ming Zhu, "Face recognition in real-world surveillance videos with deep learning method," 2017 2nd International Conference on Image, Vision and Computing (ICIVC), Chengdu, 2017, pp. 239-243.

[10] K. Zhang, Z. Zhang, Z. Li and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," in IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499-1503, Oct. 2016

[11] Matthew D Zeiler, Rob Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014: Computer Vision – ECCV 2014 pp 818-833.

[12] Redmon, Joseph & Farhadi, Ali. YOLOv3: An Incremental Improvement. Technical report. arXiv. 2018.

[13] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017.