Incorrect Facemask-Wearing Detection Algorithm Comparison

Ke Shi, [ks1837@georgetown.edu](mailto:ks1837@georgetown.edu)

Yixuan Cheng, [yc912@georgetown.edu](mailto:yc912@georgetown.edu)

Yuquan Xu, [yx241@georgetown.edu](mailto:yx241@georgetown.edu)

Data Science and Analytics, Graduate School of Arts and Sciences,

Georgetown University, Washington DC

# Introduction

The COVID-19 pandemic has wreaked havoc all over the planet. The pandemic has brought profound change to people’s economic and social lives. During this global health crisis, numerous measures were enforced to fight off the adverse effects of this pandemic. Wearing facemasks has become one of the most influential and essential mechanisms governments adopt worldwide. The correct use of masks necessitates careful attention to general hygiene standards. The most important is adequate mouth and nose coverage while avoiding gaps between the face and the mask. However, a large number of people are reluctant to put on masks correctly, which poses a high risk for the transmission of infection. Thus, it is crucial to control the incorrect use of masks and guide people to wear masks correctly as they enter public spaces.

The primary purpose of our study is to compare the performance and accuracy of different methods proposed to distinguish whether a person is wearing a mask properly. We aim to determine which algorithm yields the highest accuracy to help determine the best practice to incorporate. Traditionally, face mask detection was applied to prevent terrorist attacks and reduce illegal and criminal conduct (Ejaz et al., 2019, Ge et al., 2017). Following the COVID-19 epidemic, researchers conducted several face mask occlusion studies (Su et al., 2021). YOLO (Redmon et al., 2016), VGG (Simonyan & Zisserman, 2015), Inception V3 (Szegedy et al., 2015), ResNet (He et al., 2016), ReLU (Agarap, 2020), and various classification systems were used to detect if the mask was worn correctly.

# Background/Review of the Literature

Novel strategies to detect the inappropriate use of preventative mechanisms are a heated topic because of the substantial risk of a COVID-19 pandemic. Deep Learning, specifically Convolutional Neural Networks (CNN) are prime method utilized to handle image categorization and pattern detection tasks (Tomás et al., 2021). Some applications were already being developed to control the use of masks during other pandemics. However, in the last two years, the quantity of publications related to these classification systems has risen drastically due to our serious position. (Tomás et al., 2021).

Nagrath et al. (2021) created a device that can tell if a face mask is being used or not. They employed the MobileNetV2 architecture (Howard et al., 2017) as the foundation for the classifier, and the Single Shot MultiBox Detector MobileNetV2 (SSDMNV2) technique, which employs a single-shot multibox detector as a face detector, to achieve real-time mask detection. They suggest using deep learning TensorFlow, Keras, and OpenCV to identify face masks using these methods. This study has an accuracy of 92.64 percent and an F1 score of 0.93. Mata (2021) developed a CNN model to distinguish between those who wear masks and those who do not. It is based on a deep learning algorithm that uses an image or video stream as input.

Nieto-Rodríguez et al., (2015) built a real-time image processing system with VGA resolution and a10 frames per second frame rate. Faces or masks may be distinguished up to 5 meters away with VGA resolution. The system was then utilized to regulate medical personnel’s use of masks in operating rooms. An alarm will sound if the health staff fails to follow the direction (Tomás et al., 2021).

Since wearing masks incorrectly greatly diminishes the effectiveness against the virus, several researchers have concentrated on recognizing its accurate or wrong placement in addition to detecting its existence or absence (Tomás et al., 2021). Rudraraju et al. (2020) designed a two-step application. In the first step, the application was designed to detect if a facial mask was being worn or not. The second step determines whether it was being used correctly or incorrectly. Authors utilized fog computing to perform this task. The video sequence is processed by two nodes. Two MobileNet models are implemented in each fog node (Howard et al., 2017). Haar cascade classifiers are used to recognize faces. Without relying on the Internet, streaming happens locally at each fog gateway. Only the mask is permitted to enter the room in this manner, and only if the mask is installed correctly. This method has a 90 percent accuracy rate.

Using hybrid machine learning approaches, Wang et al. (2021) presented a two-stage methodology to identify the use of masks. The user wearing a face mask is identified in the first stage utilizing the Faster RCNN and InceptionV2 (Szegedy et al., 2016) structural model. The second phase involves a stage of verification of genuine face masks using a learning system and a classifier. The overall accuracy for basic scenarios is 97.32 percent, whereas it is 91.13 percent for more difficult scenarios.

Although numerous related articles have shed light on the algorithm to identify whether the mask is worn properly, there are limited pieces of literature dedicated to recreating and comparing the preexisting detection models. Our work is dedicated to filling the void by duplicating the feature detection models, performing analysis on the same dataset, and drawing conclusions on which model performs best.

# Data Gathering and Preliminary Methods Review

## Research Questions - Answer

Question: Based on the photos in the database as the training set and validation set, using five different algorithms to identify which person in the photos has placed the mask in the correct place and which is in the incorrect place, and how’s four algorithms’ accuracy shown on our test set(own taking pictures)

Answer: Each person in each photo will be given a label, without the mask, KN95 mask, N95 mask, and surgical mask. We are using YOLO V4 (Bochkovskiy, 2020), ResNet (He, 2015), ReLU (Agarap, 2020), and Inception 3 (Szegedy, 2015) four different algorithms to train the training set. Then uses these training results to predict our test set and decide which algorithm is the best for this process.

## Preliminary Methods

Currently four methods are being selected for this project, which are YOLO V4 (Bochkovskiy, 2020), ResNet (He, 2015), ReLU (Agarap, 2020), Inception 3 (Szegedy, 2015). These four methods are all frequently used and have very good performance in related fields. For example, yolov4 was proposed in 2020 and is a more advanced version of Yolov3, which was chosen as one of the methods for this project. All four methods will be described in more detail in the following paragraphs.

## Dataset

Two or more datasets will be selected as the data for this project. Since facial photos may involve personal privacy, we do not consider using unauthorized photos on social media at this time. The reason we chose to have more than one dataset is that the images and tones in the same dataset may be extremely similar, and different datasets may greatly reduce similar problems. We also want to investigate in a follow-up study how the performance of different methods differs in different datasets and find the reasons for this.

All data are pre-processed in the same way and the data set is divided into training and test sets in a 4:1 ratio.The first dataset will be publicly available on "Paddle Paddle AI Studio" (Xiao), which includes photos of five volunteers wearing different styles of masks, such as KN95, N95. Second one will also be available on "Paddle Paddle AI Studio" (2021). The dataset collected 500 images of faces without masks and simulated 1000 simulated images with masks based on these images. The third dataset that is being select is Real-world Masked Face Recognition Dataset (RMFRD). (Wang et al., 2020). It is a more comprehensive dataset contains over 90000 photographs.

# Empirical Design

## Data Description

The primary goal of the research is to detect whether the object is wearing the masks correctly and compare the accuracy of different algorithms on the same dataset. In order to test the existing feature detection models to their fullest capacities, the dataset chosen needs to contain as much variance as possible. We obtained our dataset from four different sources, of which three are open-source data, and one is self-obtained. Our dataset contains images with simulated and real masks taken from various angels in assorted lighting conditions with differing resolutions.

The first dataset is publicly available on "Paddle Paddle AI Studio" (Xiao). It contains 500 sets of images with Simulated Masked Face from 5 volunteers wearing KN95, N95, and surgical masks, respectively (7500 images in total). The images are cropped to follow a uniform dimension, 160 x 160 pixels. An accurate masked face detection model that can distinguish among different common types of masks could be trained using this dataset. In order to expand the volume and diversity of the dataset for better training of the networks to recognize masks (Wang et al., 2020), we employed a second dataset which is also publicly accessible through Paddle Paddle AI Studio (2021). The dataset consisted of images of various dimensions from 500 people without masks and with the simulated mask as comparison groups crawled from the internet (1000 images in total). The third dataset utilized in this study is Real-world Masked Face Recognition Dataset (RMFRD), created by Wang et al. (Wang et al., 2020) from Wuhan University. From vast Internet resources, public figure front-face photos and their related masked face images are crawled using a Python crawler program to create RMFRD. The inappropriate face pictures caused by incorrect correspondence are already carefully removed. Moreover, the researchers also adopt semi-automatic annotation tools like LabelImg (Lin, 2015) and LabelMe (Russell et al., 2007) to crop the precise facial regions. The dataset contains 90,000 photographs of 525 participants without masks and 5,000 shots of the same 525 subjects with masks. (Wang et al., 2020). The simulated masked face datasets can be utilized in conjunction with their authentic unmasked counterparts to enhance the ability of the networks to differentiate between two kinds of masks. We also incorporated the 100 images of the group member, family, and friends wearing masks properly and improperly to complete the dataset further.

## Dependent Variable/Feature/object

The purpose of our analysis is to reproduce the four existing object detection neural network: YOLO V4 (Bochkovskiy, 2020), ResNet (He, 2015), ReLU (Agarap, 2020), and Inception 3 (Szegedy, 2015), perform analysis on the same dataset we have to collect to detect whether the subject is wearing masks incorrectly and draw a conclusion on which algorithm has the best performance in terms of accuracy and speed. There are three image categories: no mask, adequately worn masks, and improperly worn masks. For the images with incorrectly worn masks, we will further label them manually by their different characteristics. The labels are 1- nose exposure, 2- without masks, 3- incorrectly bent in the nasal part, 4 - glasses placed under the mask, 5- incorrectly placed rubber band, and 6- others. (Tomás, 2021) Despite the many more wrong ways to wear a mask, we would only investigate and detect the relatively more obvious incorrectness due to the time constraint.

## Data Preprocessing

The images in the Real-world Masked Face Recognition Dataset have already been preprocessed by removing the inappropriate face pictures caused by incorrect correspondence and cropping the precise facial regions with the help of semi-automatic annotation tools like LabelImg (Lin, 2015) and LabelMe (Russell et al., 2007). (Wang et al., 2020). Moreover, the dataset from Paddle Paddle AI studio was also cropped to follow a uniform dimension, 160 x 160 pixels. However, not all the images in our dataset are ready for analysis and object detection. Different algorithms require different dimensions of the image input. For example, the ResNet-50 network can accept input images with height and width multiples of 32. (Sachan, 2019) We set the standard size as 224 x 224. Since the image selected are all unwrapped and in RGB color space. The only preprocessing step we need to do is crop the image to follow a uniform dimension and only emphasize the facial regions.

Graphical user interface, application

Description automatically generated

## Construction of Training and Testing Set

The validation data are coming from the online face mask dataset since we cannot invent an app that let people take photos or use the camera to capture people’s faces. The first dataset we are using is the common dataset which is used for training neural networks. The second dataset we are using is designed to distinguish people with masks or not. The third dataset is photos that contain different face sizes and weather conditions, which are more combined with reality. The dataset we find is a qualified dataset that has been used by thousands of people, which makes the dataset more compliant. The way we imitate is in the first paper below. They also download the data from two online resources, which removed the dataset bias. These images from the online dataset will be used as a training set and validation set with ratios of 0.8 and 0.2 for both two datasets. For the test set, our own group members and friends will take pictures using our phones or cameras to imitate real-life situations. We want to have more realistic and close-to-life results using different algorithms.

The first article (Su, 2021) which is in the link directly uses the RMFD dataset and MAFA dataset shown on the website, the dataset has already been validated and will only contain people's faces with the mask or not. The mask dataset has already been prepared for them. The authors divide the datasets into a training set, validation set, and test set with ratios of 0.7, 0.1, and 0.2. They are using two datasets as mixed datasets. The training set and the test set will contain pictures from both datasets. Since they cannot get their own datasets, such as taking photos on their own or getting photos from monitors. The only way they can do this is to split the datasets to make the explanation seem more reasonable. However, since the dataset is provided by the organization. The algorithms they provided may only be worked for specific datasets. The result showed that the deep learning method has different results for the different datasets. The algorithm they are using is not very persuasive for people in daily life. We still need to decide which algorithm is best for different datasets.

The second article (Tomás, 2021) uses a different method for the validation dataset. They developed a real-time demonstration on mobile, and the app can be downloaded by everyone. Using the camera, the presence of a face will be detected and by using the voice message, if the facemask-wearing problem is detected, the user will be thanked for its correct use. However, they also have some limitations on the picture. For example, it has to be at the entrance of a public place. Since this paper focuses on the top five incorrect face mask problems, they only choose the images that only present one type of problem or none. These images have been divided into two sets, 1000 for training and 194 for validation. The focus of their research is not on whether this person put on a mask or not, it’s if the mask is correct or not. The dataset they are chosen from are the user-provided images in their user library. The way they get the images are great, but it’s hard for us to imitate. However, the way they are used for training and validation is not good as in the first paper. They didn’t have a specific proportion for these datasets and no test set. The division only works for its own dataset which has a lot of limitations. If it is possible, dividing the dataset into a different proportion will be better, such as the test set, train set, and validation set the ratio of 0.2, 0.7, and 0.1 might be better for this research.

## Deep Learning Method

The four algorithms we want to use are YOLO V4 (Bochkovskiy, 2020), ResNet (He, 2015), ReLU (Agarap, 2020), and Inception 3 (Szegedy, 2015). The reason for YOLO since it’s a very popular algorithm, which has already been developed four times. A lot of papers are using YOLO V3 as a base method. Therefore, we want to use a better-developed method YOLO V4 in this project. Combining with YOLO, they also use ResNet for feature detection and decision trees. The ReLU has also been mentioned in a lot of papers for its dimensionality reduction. Last but not least, Inception 3 also provided a very good result in some papers. The accuracy is very high for mask detection. We want to include inception 3 also. For each algorithm, we will use a short paragraph for summarizing this method, which will show below:

Table 1 Deep Learning Architectures

| **Architecture** | **No. of layers** | **Short description** |
| --- | --- | --- |
| AlexNet | 8 | A powerful architecture for any object-detection task and is capable of getting excellent accuracy on rather complex datasets. (Wei, 2020) |
| Inception V3 | 42 | It is the third edition of Google's Inception Convolutional Neural Network which allows for deeper networks while simultaneously limiting the amount of parameters. (Szegedy et al., 2015) |
| ReLU |  | It is a piecewise linear function that produces zero if the input is negative and directly if the input is positive. It is frequently used in neural networks as a default activation function, which helps the model run and train more effectively. (Ohri, 2022) |
| VGG | 11 | The network utilizes small 3 x 3 convolution filters. This network is characterized by simplicity. (Simonyan & Zisserman, 2015) |
| YOLO V4 | 32 | It is a one-stage object detection model that was built upon the previous model of YOLO, designed by Joseph Redmon. ((Bochkovskiy et al., 2020) |
| ResNet | 34 | ResNet is an artificial neural network that is a gateless or open-gated variant of the HighwayNet. It is the most frequently mentioned neural network of the twenty-first and won the ImageNet 2015 competition. (2022) |
| ShuffleNet | 50 | An architecture, developed by Megvii Inc. and marketed as an exceptionally computation-efficient CNN architecture for mobile devices with 10-150 MFLOPs of computing power. (Synced, 2017) |
| EfficientNet | 237 | A convolutional neural network design and scaling technique that uses a compound coefficient to consistently scale all depth, width, and resolution parameters. (Tan & Le, 2020) |
| MobileNet | 28 | A lightweight deep neural network that uses depthwise separable convolutions. It greatly decreases the number of parameters, comparing to a network with normal convolutions of the same depth in the nets. (Pujara, 2020) |

Joseph Redmon, who is currently retired from CV, created the first YOLO (You Only Look Once) using a unique framework called Darknet. The greatest real-time object detectors in computer vision have been created by Darknet, a very flexible research framework implemented in low level languages, including YOLO, YOLOv2, YOLOv3, and recently YOLOv4. (Solawetz, 2022) A modern detector typically consists of two parts: a head that predicts object classes and bounding boxes and a backbone that is pre-trained on ImageNet. The layers that are often inserted between the backbone and the head by object detectors created in recent years are typically utilized to gather feature maps at various levels. It might be referred to as an object detector's neck. (Bochkovskiy et al., 2020) YOLOv4 used CSPDarknet53(Wang et al., 2020) as the backbone, Spatial Pyramid Pooling (SPP) (He et al., 2015) and Path Aggregation Network (Liu et al., 2018) as the neck, and YOLOv3(Redmon & Farhadi, 2018) as the head. In terms of both speed and accuracy among all the available alternative detectors, YOLOv4 presents with the best performance. (Bochkovskiy et al., 2020)

ResNet, which is short for Residual Network was firstly proposed by Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun in 2015. The main idea of ResNet is “identity shortcut connect”, which means skipping some of the layers without extra parameters nor computational complexity (He, 2015). By doing so, Resnet has a very good performance as well as high accurate rate, which Won many prizes such as 1st place in the ILSVRC 2015 classification competition with a top-5 error rate of 3.57%.

Rectified Linear Units (ReLU) was developed by Agarap in 2020. It is a classification function in a deep neural network, which is used as an activation function with the Softmax function. ReLU is accomplished by taking the activation of the penultimate layer h\_(n-1) in a neural network, then multiplying it by weight parameter θ to get the raw scores o\_i by 0, i.e. f(o) = max(0, o\_i), where f(o) is the ReLU function. (p. 1)

Inception 3 is developed by the increased model size and computational cost. Computational efficiency and low parameter count are enabling factors for various use cases such as mobile vision and big data scenarios. Scaling up networks in ways that aim at utilizing the added computation as efficiently as possible by suitably factorized convolutions and aggressive regularization. Based on ILSVRC 2012 classification challenge validation set, it demonstrates 21.2% top-1 and 5.6% top-5 error for single frame evaluation using a network with a computational cost of 5 billion multiply-adds per inference and with using less than 25 million parameters. With an ensemble of 4 models and multi-crop evaluation, it has a 3.5% top-5 error and 17.3% top-1 error. (Szegedy, p. 1)

# Experiment and Results

In the following analysis, different models will be applied for each test and create a table that we will call Table 1 (Table 1 is the average result of all data sets + all models + all metrics). In order to better measure the computational accuracy of different models in multiple aspects, Table 1 uses three metrics for the dataset and records the mean value of multiple datasets to measure the efficiency of the models together.

To have an objective evaluation, all models are trained and tested in the same hardware environment, and all hardware is most-advanced GPU and CPU in order to ensure less influence by hardware.

## Performance Comparison of Different Models

We used five metrics to analyze the performance of the model from several aspects. The five metrics are recall, precision, MRAE, RMSE, and the time taken to compute every 100 images. We set an IoU threshold of 0.5, which is the ratio between the intersection and the concatenation of the actual target region and the predicted target region. When the IoU of the prediction result is greater than the threshold value, the region is considered to have a target. Based on this, the confusion matrix is made.

|  |  |
| --- | --- |
| TP | FP |
| TN | FN |

Based on the confusion matrix, recall and precision can be calculated.





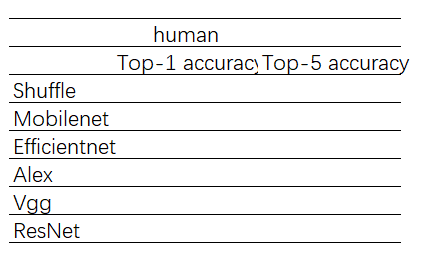
Recall represents the coverage rate of our prediction results, the higher this value means the more complete our prediction results are. However sometimes the model tends to identify a large number of wrong targets as correct targets in order to cover as many actual targets as possible. However, the model tends to miss a large number of targets in order to increase the precision rate as much as possible. Therefore, a combination of both is needed to compare whether a model is able to predict fully and accurately. Also, we will measure the performance of the model using two value, MRAE and RMSE (Arad, 2020).

Diagram

Description automatically generated

MRAE represents the relative error of the regression position, which is used to measure the model performance when the target is small. RMSE represents the absolute error of the regression position, which is used to measure the model performance when the target is large.

Time represents the average time consumed by the model per 100 plots computed and is used to measure the real-time performance of the model. We will record the results of the experiment in table1.

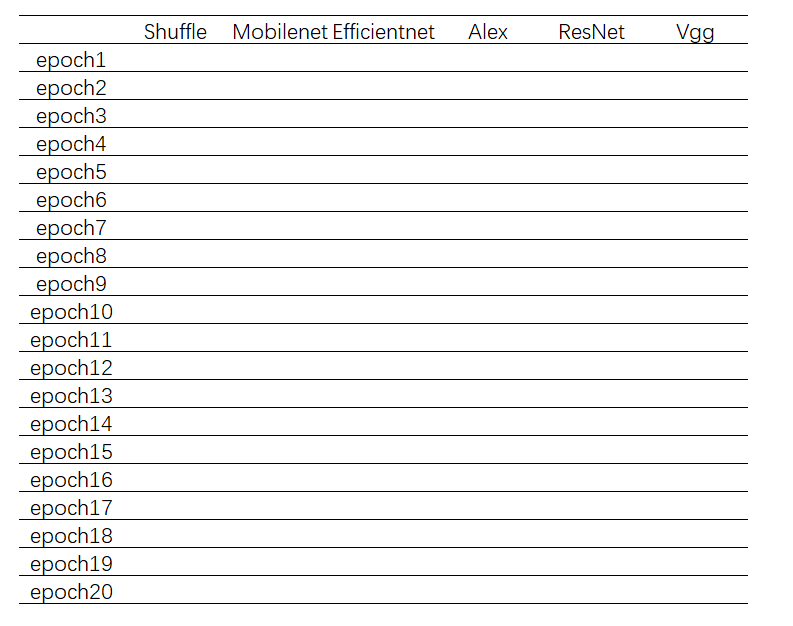


Through the table we can collect the performance differences of different models on different datasets.

## Studying the effect of hyperparameters on accuracy

The hyperparameters of deep learning have a great influence on the prediction results, mainly three parameters are model depth, generation and batch size, and the effects of different hyperparameters on the accuracy of the training set and the accuracy of the test set are recorded, as shown in Table 2.

Train Loss



Model depth is mainly expressed in the number of layers of feature extraction, when the model is shallow, the model can extract the spatial features of objects or targets, and as the depth increases, the model is gradually able to extract the semantic features of the target, which represents that the model can understand what the target is more abstractly without misjudging with the change of the scene. Batch size represents the number of images read in each time. Epoch represents the number of times a training set is trained, different training times can affect the final result, but the more epochs are not always better. Generally, for different datasets, the epochs with the best results are often different.

## Performance on different dataset sizes

The larger number of datasets with not wearing masks, wearing masks and not properly wearing masks will be applied to calculate the training and testing accuracy of different models. Due to the different focus of the deep learning network, as the size of the same dataset increases, the features extracted by some models lose their relevance in different images, which will lead to the occurrence of overfitting in the small sample space and underfitting in the large sample space. Eventually, the training data set becomes larger but the accuracy decreases.

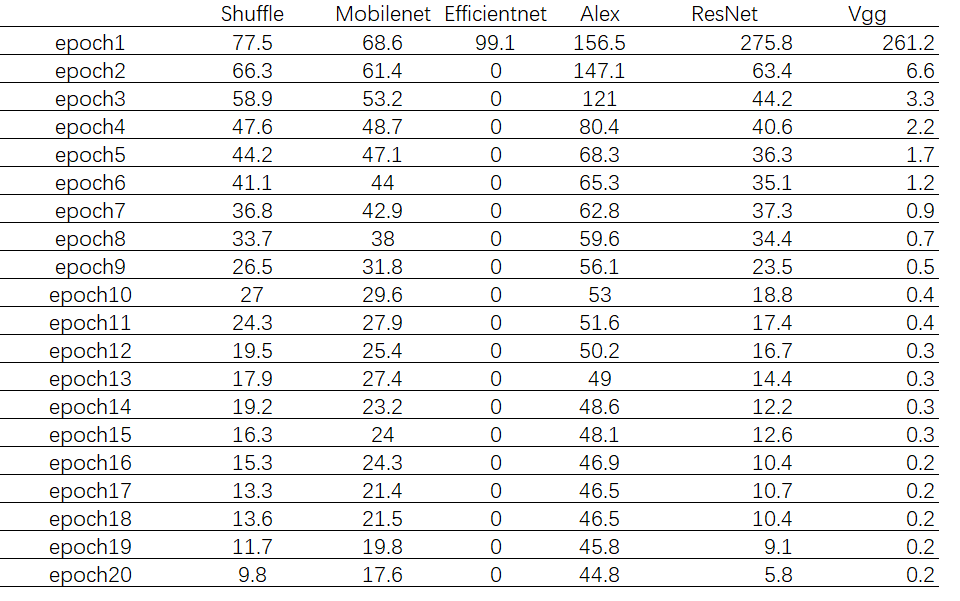
# Experiment and Results

## Task1: Determine Whether There is Presence of Huma in the Image

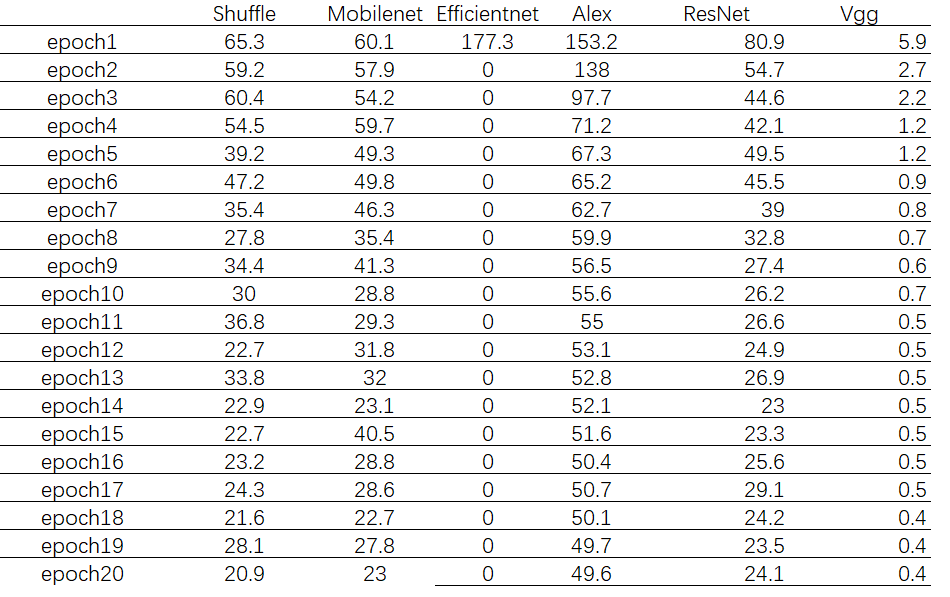
Chart, bar chart

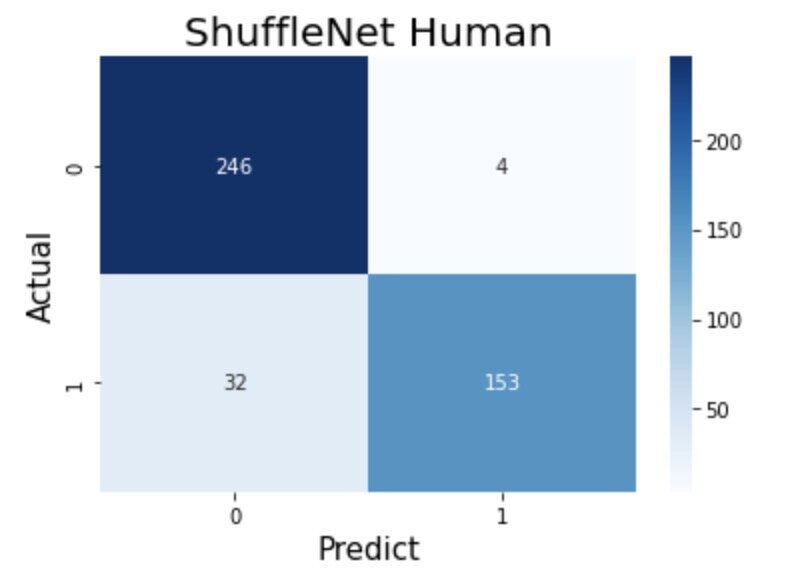
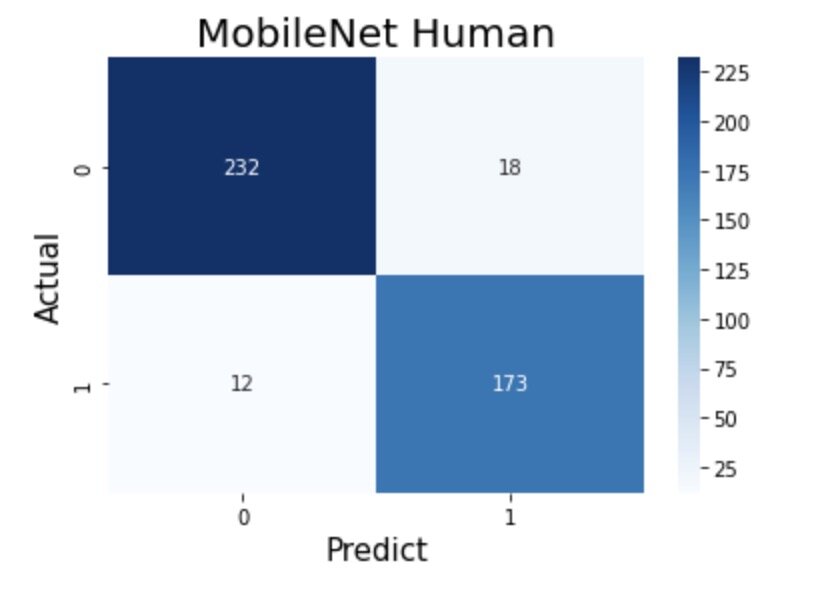
Description automatically generated

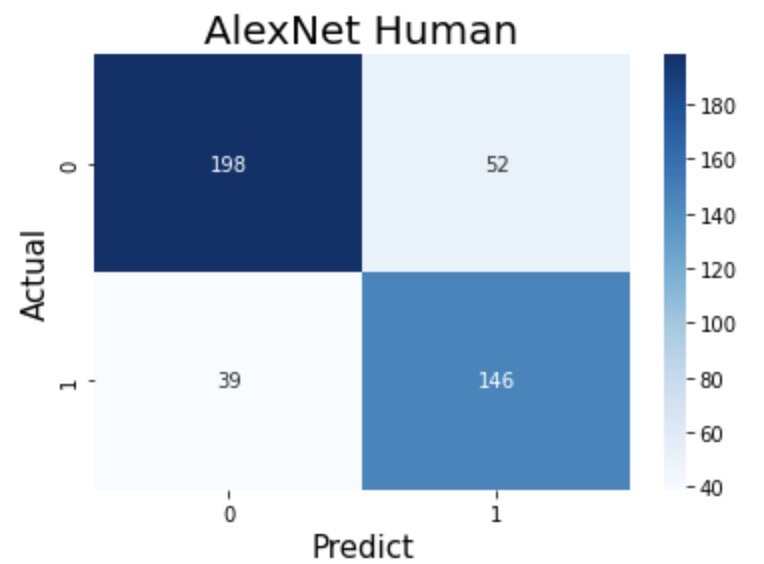
Train Loss

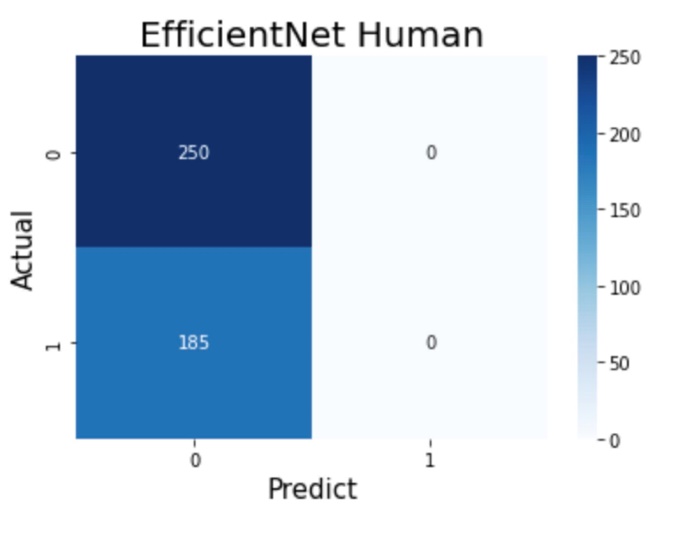
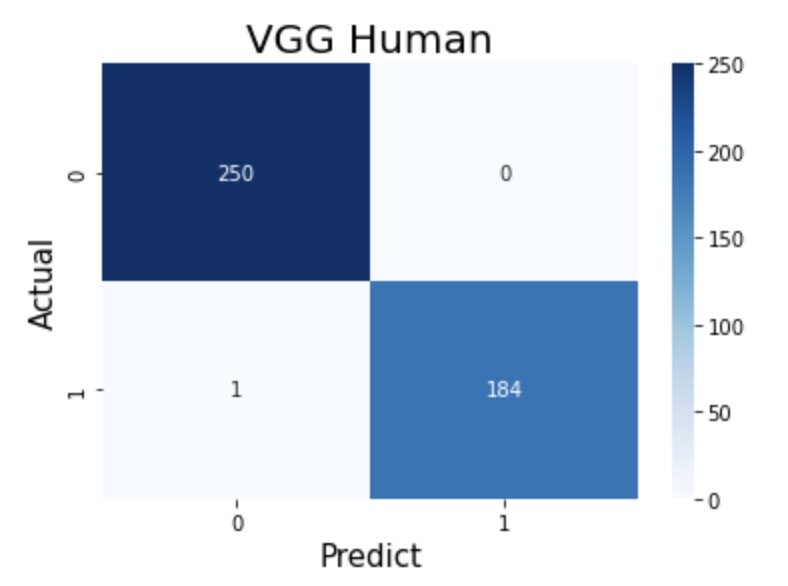


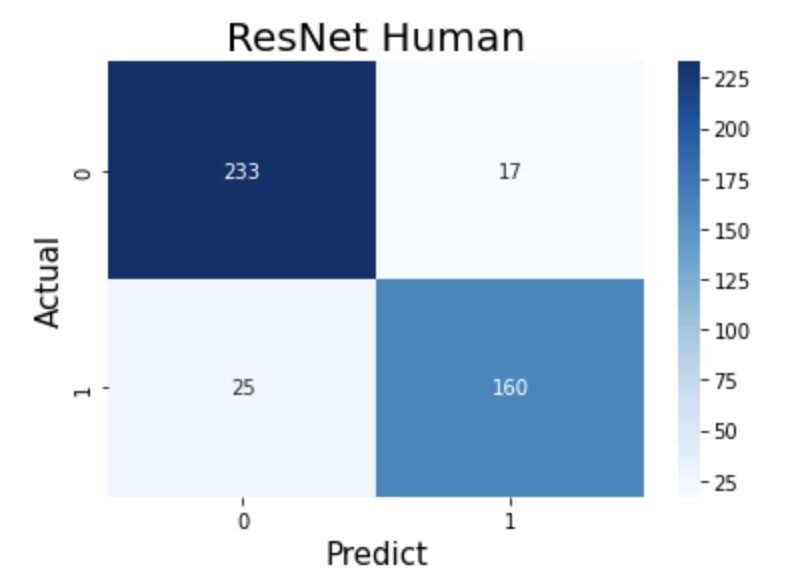
Test Loss



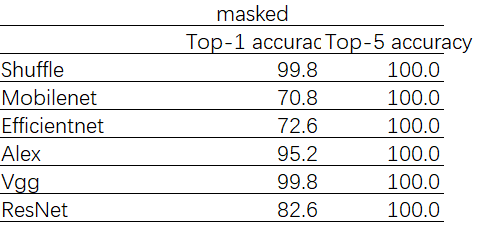








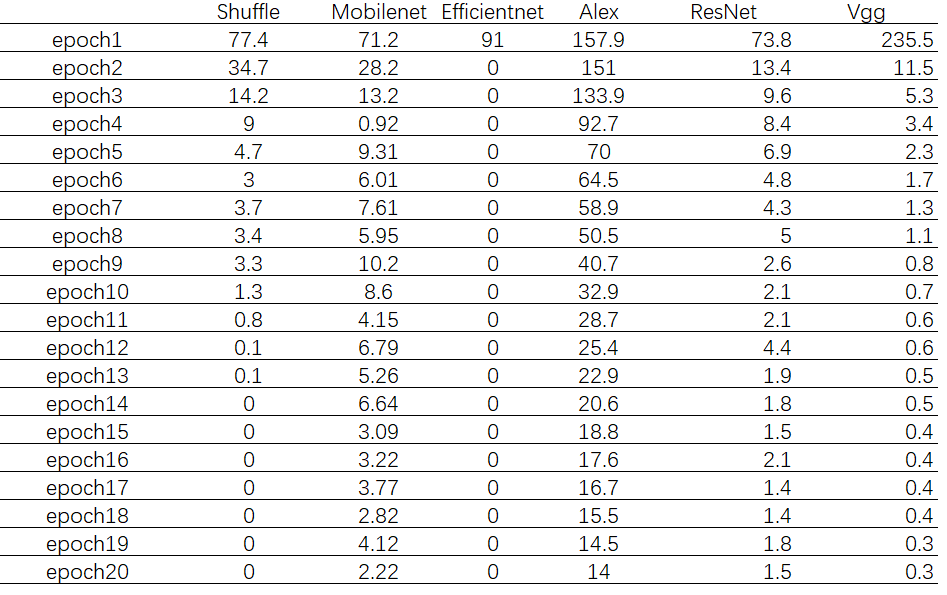
## Task2: Determine the Presence of Mask in the Image

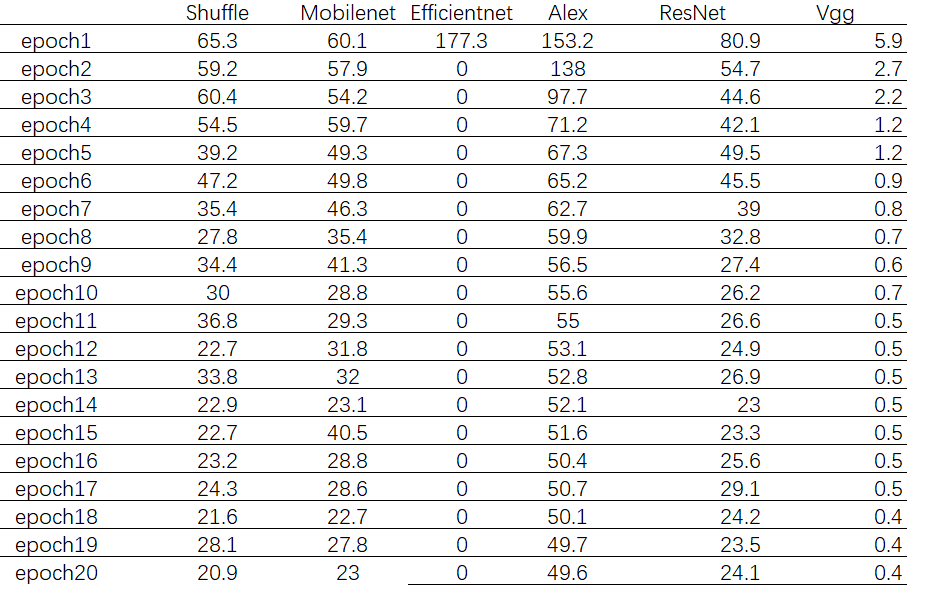


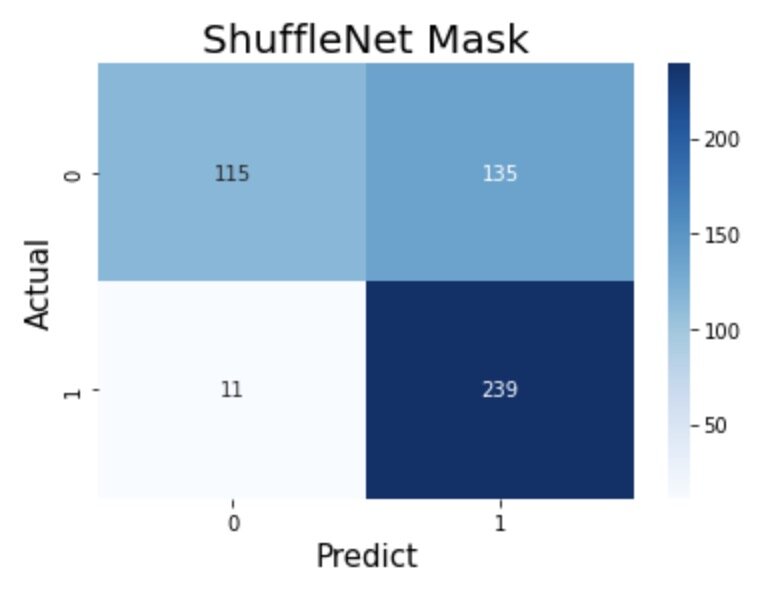
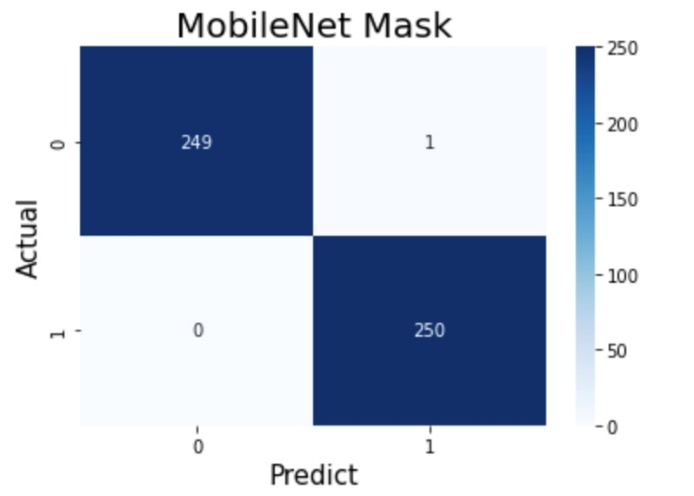
A picture containing text, clipart

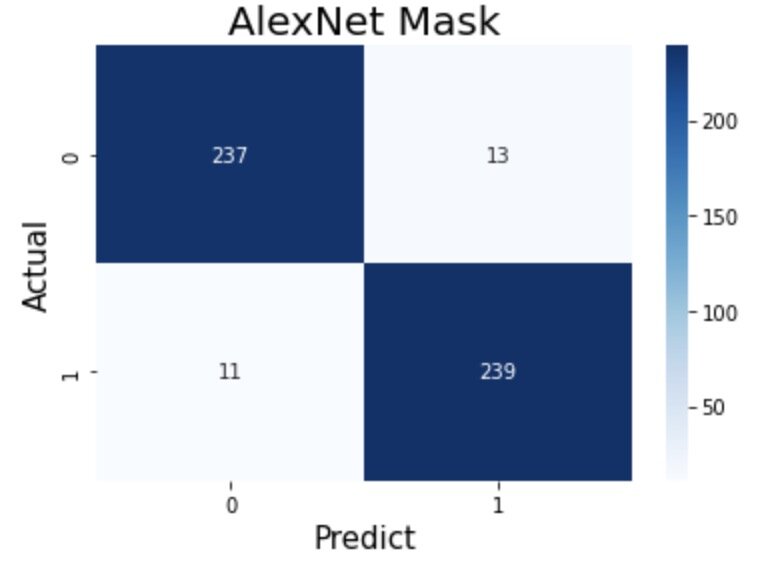
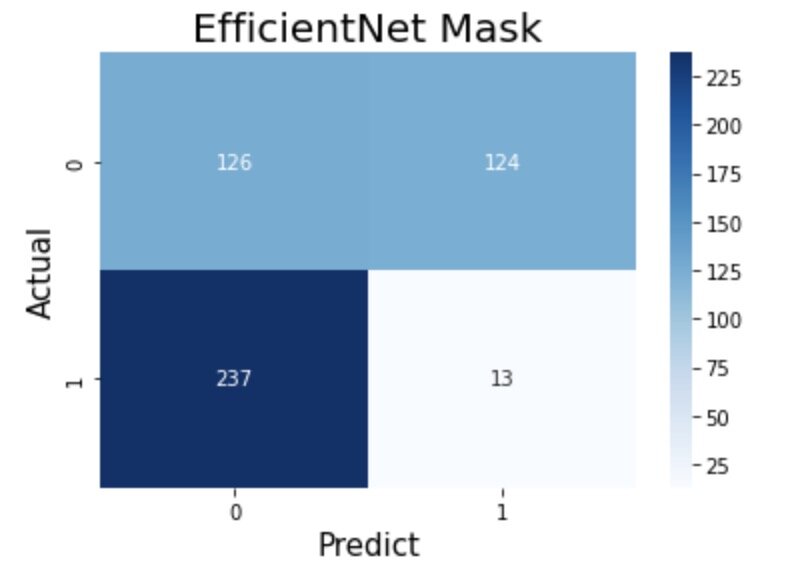
Description automatically generated

**TRAIN LOSS**









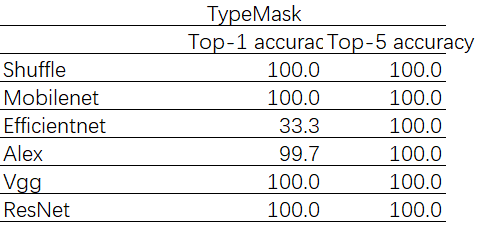
Chart

Description automatically generatedChart

Description automatically generated

## Task3: Distinguish Types of Masks in the Image

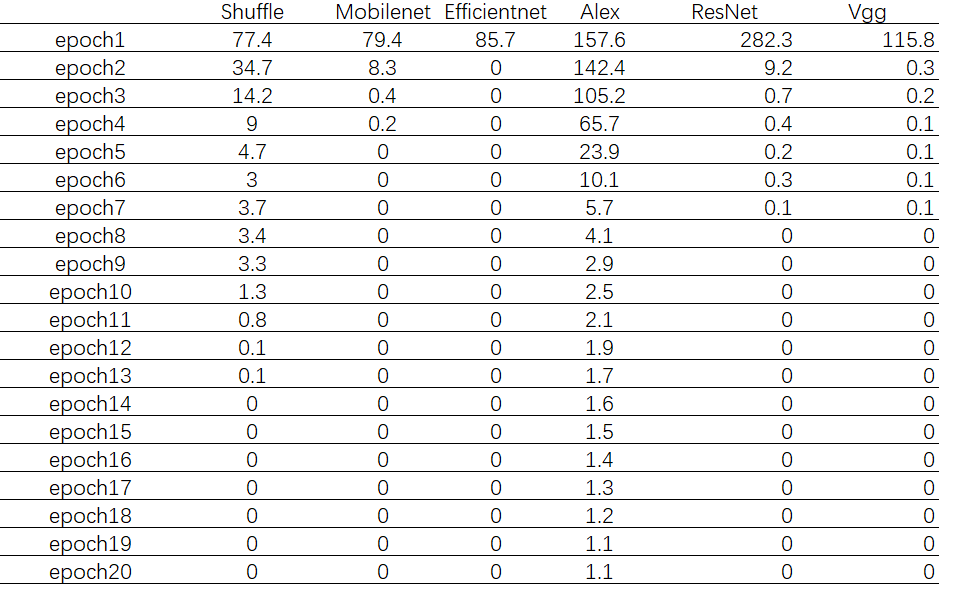
Three types of masks are presented in the dataset including N95, KN95, and Surgical.



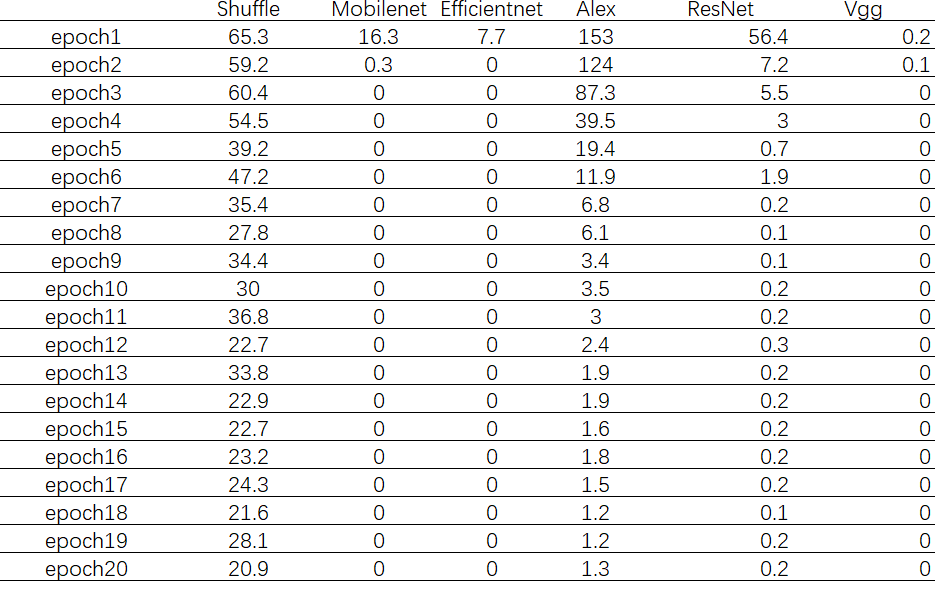
A picture containing text

Description automatically generated

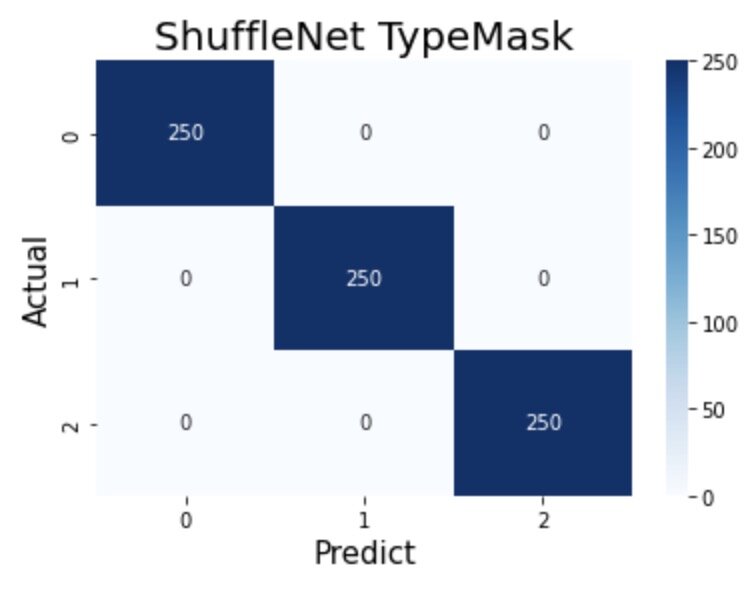
TRAIN LOSS

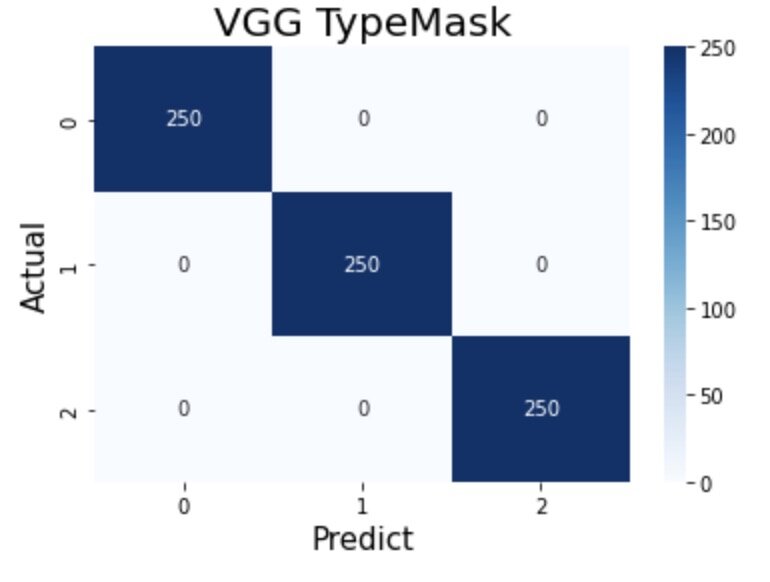
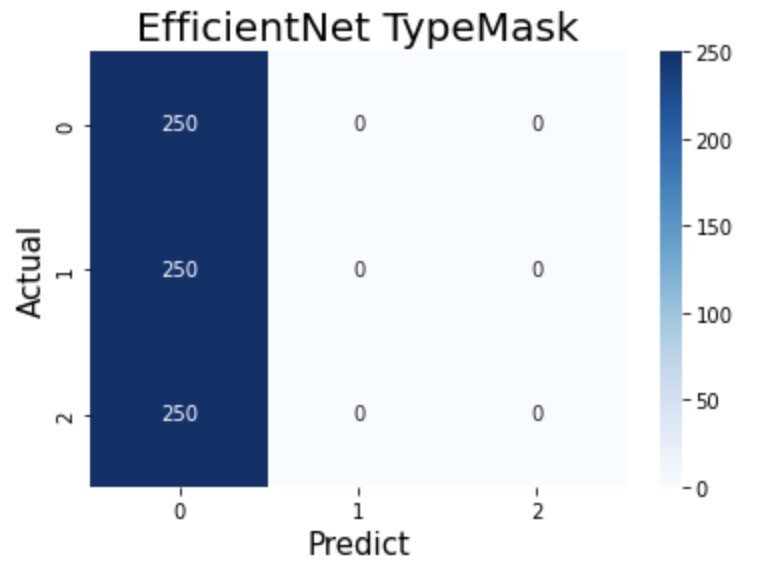
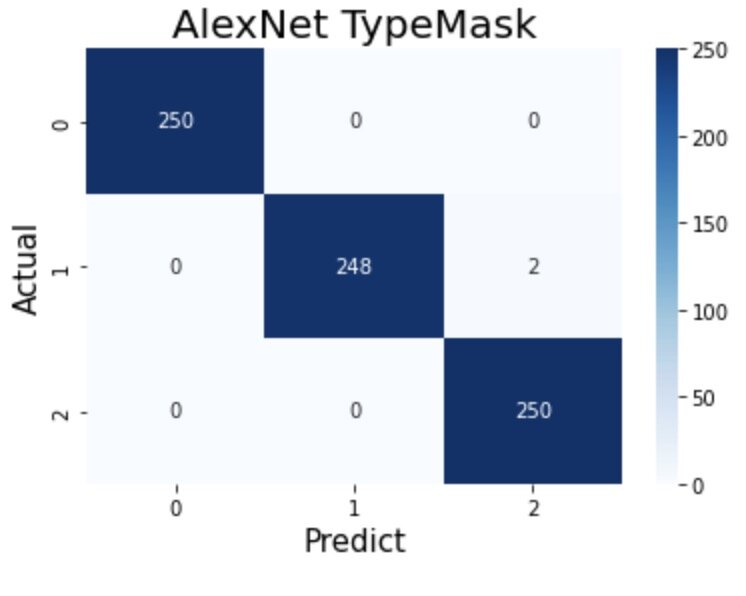
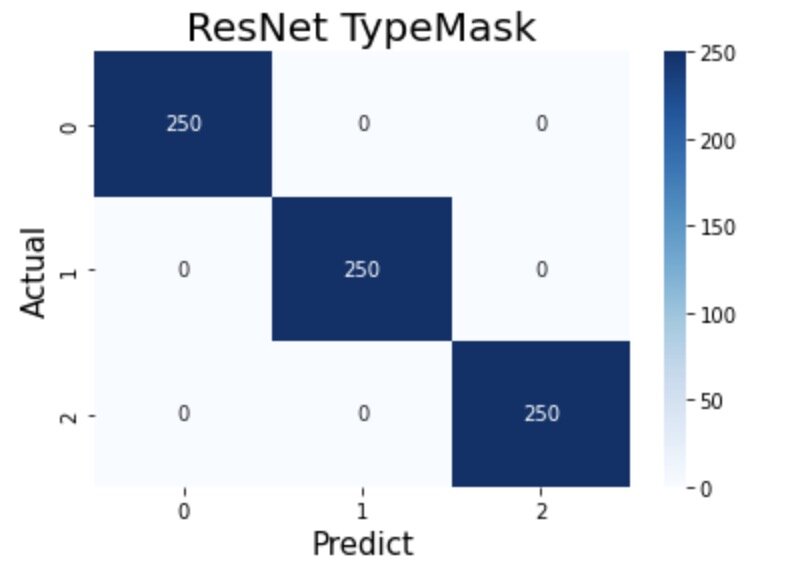


TEST LOSS



Chart

Description automatically generated



# Conclusion

According to the experimental results, since the number of classifications is within 5, the Top-5 accuracy are 100% for all tasks. For different classification tasks, when the training epoch is within 20 times, VggNet has the best performance in distinguishing human and non-human images, ShuffleNet and VggNet both have good results in distinguishing images with or without masks, and for the classification task of wearing different masks, all the five methods, except EfficientNet have good performance. Therefore, VggNet can have the best performance in all three classification tasks: human and non-human, wearing or not wearing a mask, and wearing different masks, while EfficientNet has difficulty in obtaining a usable result with fewer iterations due to its overly complex network structure. The performance of ShuffleNet is similar to VggNet, with only a slight lack of performance when differentiating between human and non-human. In terms of convergence during training, ResNet, VggNet and MobileNet converge quickly, converging in the first ten epochs with good results, while ShuffleNet has better results. However, in the task of distinguishing whether to wear a mask or not, the convergence is unstable and requires more iterations and longer training time. Looking at the differences between different kinds of images in the three tasks, the differences between human and non-human images are the largest in task 1, the smallest in task 3 for images with different masks, and the smallest in task 2. According to the experimental results, the overall accuracy of these models is the lowest in task 1 and the highest in task 3, which indicates that these models may be suitable for inter-image classification tasks that produce differences in a smaller range

# Extra Question (not for now):

Q1.1: In the Object Recognition part, is there any correlation between the pixels of the mask and pixels of the person’s face, will it influence the accuracy of each algorithm?

Explain: When detecting photos, using the algorithms with different numbers of pixels and finding how many pixels will give the best accuracy and how many pixels will have the most timesaving. Considering the time and accuracy, the best algorithm will be chosen.

Q1.2: In the Feature Detection part, which will be the key factor to get the most important feature in determining the accuracy of algorithms?

Explain: Test photos will be the selfies that group members take, these test photos will include multi-angled face pictures, different backgrounds, different size photos, and different resolution pictures. Using these different types of photos as a sample, which algorithm will have the best performance in the project.

Q1.3: Given the limited resources, time, and sample pictures, which algorithm will provide the most reasonable answer for now?

Explain: Processing photos and getting each feature detected will be very time-consuming. The goal is to balance the time-consuming and reasonable accuracy of checking if people wear a mask or not.

# Reference List

1. Arad, B., Timofte, R., Ben-Shahar, O., Lin, Y. T., & Finlayson, G. D. (2020). Ntire 2020 challenge on spectral reconstruction from an rgb image. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (pp. 446-447).
2. Brownlee, J. (2020, August 20). A Gentle Introduction to the Rectified Linear Unit (ReLU). Machine Learning Mastery.<https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/>
3. Redmon, J. (2018). YOLO: Real-Time Object Detection. Https://Pjreddie.Com/Darknet/Yolo/.<https://pjreddie.com/darknet/yolo/>
4. Su, X. (2021, December 9). Face mask detection and classification via deep transfer learning. SpringerLink. https://link.springer.com/article/10.1007/s11042-021-11772-5?error=cookies\_not\_supported&code=47f7ea14-4521-4687-bcb2-4cd725cd9c26#Abs1
5. Szegedy, C. (2015, December 2). Rethinking the Inception Architecture for Computer Vision. arXiv.Org.<https://arxiv.org/abs/1512.00567v3>
6. Wang, Z., Wang, G., Huang, B., Xiong, Z., Hong, Q., Wu, H., Yi, P., Jiang, K., Wang, N., Pei, Y., Chen, H., Miao, Y., Huang, Z., & Liang, J. (2020, March 23). *Masked face recognition dataset and application*. arXiv.org. Retrieved June 22, 2022, from https://arxiv.org/abs/2003.09093
7. Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2007). LabelMe: A database and web-based tool for image annotation. *International Journal of Computer Vision*, *77*(1-3), 157–173. https://doi.org/10.1007/s11263-007-0090-8
8. Ejaz MS, Islam MR, Sifatullah M, Sarker A (2019) Implementation of principal component analysis on masked and non-masked face recognition. In: 2019 1st International conference on advances in science, engineering and robotics technology (ICASERT). IEEE, pp 1–5
9. Ge S, Li J, Ye Q, Luo Z (2017) Detecting masked faces in the wild with lle-cnns. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2682–2690
10. Tomás, J., Rego, A., Viciano-Tudela, S., & Lloret, J. (2021). Incorrect Facemask-Wearing Detection Using Convolutional Neural Networks with Transfer Learning. Healthcare (Basel, Switzerland), 9(8), 1050. <https://doi.org/10.3390/healthcare9081050>
11. Nieto-Rodríguez A., Mucientes M., Brea V.M.  *Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis, Santiago de Compostela, Spain, 17–19 July 2015.* Springer; Cham, Switzerland: 2015. System for medical mask detection in the operating room through facial attributes; pp. 138–145. [[Google Scholar](https://scholar.google.com/scholar_lookup?title=Proceedings+of+the+Iberian+Conference+on+Pattern+Recognition+and+Image+Analysis,+Santiago+de+Compostela,+Spain,+17%E2%80%9319+July+2015&author=A.+Nieto-Rodr%C3%ADguez&author=M.+Mucientes&author=V.M.+Brea&publication_year=2015&)]
12. Nagrath P., Jain R., Madan A., Arora R., Kataria P., Hemanth J. SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2. *Sustain. Cities Soc.* 2021;66:102692. doi: 10.1016/j.scs.2020.102692. [[PMC free article](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7775036/)] [[PubMed](https://pubmed.ncbi.nlm.nih.gov/33425664)] [[CrossRef](https://doi.org/10.1016%2Fj.scs.2020.102692)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Sustain.+Cities+Soc.&title=SSDMNV2:+A+real+time+DNN-based+face+mask+detection+system+using+single+shot+multibox+detector+and+MobileNetV2&author=P.+Nagrath&author=R.+Jain&author=A.+Madan&author=R.+Arora&author=P.+Kataria&volume=66&publication_year=2021&pages=102692&pmid=33425664&doi=10.1016/j.scs.2020.102692&)]
13. Howard A.G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv Prepr.* 20171704.04861 [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=arXiv+Prepr.&title=Mobilenets:+Efficient+convolutional+neural+networks+for+mobile+vision+applications&author=A.G.+Howard&author=M.+Zhu&author=B.+Chen&author=D.+Kalenichenko&author=W.+Wang&publication_year=2017&)]
14. Mata B.U. Face Mask Detection Using Convolutional Neural Network. *J. Nat. Remedies.* 2021;21:14–19. [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=J.+Nat.+Remedies&title=Face+Mask+Detection+Using+Convolutional+Neural+Network&author=B.U.+Mata&volume=21&publication_year=2021&pages=14-19&)]
15. Rudraraju S.R., Suryadevara N.K., Negi A. Face Mask Detection at the Fog Computing Gateway; Proceedings of the 2020 IEEE 15th Conference on Computer Science and Information Systems (FedCSIS); Sofia, Bulgaria. 6–9 September 2020; pp. 521–524. [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Proceedings+of+the+2020+IEEE+15th+Conference+on+Computer+Science+and+Information+Systems+(FedCSIS)&title=Face+Mask+Detection+at+the+Fog+Computing+Gateway&author=S.R.+Rudraraju&author=N.K.+Suryadevara&author=A.+Negi&pages=521-524&)]
16. Wang B., Zhao Y., Chen C.P. Hybrid Transfer Learning and Broad Learning System for Wearing Mask Detection in the COVID-19 Era. *IEEE Trans. Instrum. Meas.* 2021;70:1–12. [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=IEEE+Trans.+Instrum.+Meas.&title=Hybrid+Transfer+Learning+and+Broad+Learning+System+for+Wearing+Mask+Detection+in+the+COVID-19+Era&author=B.+Wang&author=Y.+Zhao&author=C.P.+Chen&volume=70&publication_year=2021&pages=1-12&pmid=33776080&)]
17. Szegedy C., Vanhoucke V., Ioffe S., Shlens J., Wojna Z. Rethinking the Inception Architecture for Computer Vision; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); Las Vegas, NV, USA. 27–30 June 2016; pp. 2818–2826. [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Proceedings+of+the+IEEE+Conference+on+Computer+Vision+and+Pattern+Recognition+(CVPR)&title=Rethinking+the+Inception+Architecture+for+Computer+Vision&author=C.+Szegedy&author=V.+Vanhoucke&author=S.+Ioffe&author=J.+Shlens&author=Z.+Wojna&pages=2818-2826&)]
18. *口罩检测数据集 - 飞桨AI Studio*. (n.d.). Mask Detection Dataset. <https://aistudio.baidu.com/aistudio/datasetdetail/103949>
19. *口罩人脸 - 飞桨AI Studio*. (n.d.). Mask Detection Dataset. <https://aistudio.baidu.com/aistudio/datasetdetail/100447>
20. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).
21. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
22. Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2015.7298594
23. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
24. Lin, T. (2015). *Tzutalin/labelimg: LabelImg is a graphical image annotation tool and label object bounding boxes in imagesTzuta Lin*. GitHub. Retrieved June 22, 2022, from https://github.com/tzutalin/labelImg
25. Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2007). LabelMe: A database and web-based tool for image annotation. *International Journal of Computer Vision*, *77*(1-3), 157–173. https://doi.org/10.1007/s11263-007-0090-8
26. Solawetz, J. (2022, May 11). *YOLOv4 - An explanation of how it works*. Roboflow Blog. Retrieved June 22, 2022, from https://blog.roboflow.com/a-thorough-breakdown-of-yolov4/
27. Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020, April 23). *Yolov4: Optimal Speed and accuracy of object detection*. arXiv.org. Retrieved June 22, 2022, from https://arxiv.org/abs/2004.10934
28. Wang, C.-Y., Mark Liao, H.-Y., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., & Yeh, I.-H. (2020). CSPNet: A new backbone that can enhance learning capability of CNN. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. <https://doi.org/10.1109/cvprw50498.2020.00203>
29. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(9), 1904–1916. https://doi.org/10.1109/tpami.2015.2389824
30. Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path Aggregation Network for instance segmentation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr.2018.00913
31. Redmon, J., & Farhadi, A. (2018, April 8). *Yolov3: An incremental improvement*. arXiv.org. Retrieved June 22, 2022, from https://arxiv.org/abs/1804.02767
32. Sachan, A. (2019, September 17). *Detailed guide to understand and implement ResNets*. CV. Retrieved June 22, 2022, from https://cv-tricks.com/keras/understand-implement-resnets/
33. Synced. (2017, August 29). ShuffleNet: An extremely efficient convolutional neural network for mobile devices. Medium. Retrieved July 30, 2022, from <https://medium.com/syncedreview/shufflenet-an-extremely-efficient-convolutional-neural-network-for-mobile-devices-72c6f5b01651>
34. Tan, M., & Le, Q. V. (2020, September 11). EfficientNet: Rethinking model scaling for Convolutional Neural Networks. arXiv.org. Retrieved July 30, 2022, from <https://arxiv.org/abs/1905.11946v5>
35. Pujara, A. (2020, July 15). Image classification with MobileNet. Medium. Retrieved July 30, 2022, from https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470
36. Wei, J. (2020, September 25). Alexnet: The architecture that challenged CNNS. Medium. Retrieved July 30, 2022, from https://towardsdatascience.com/alexnet-the-architecture-that-challenged-cnns-e406d5297951
37. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2015.7298594
38. Simonyan, K., & Zisserman, A. (2015, April 10). Very deep convolutional networks for large-scale image recognition. arXiv.org. Retrieved July 30, 2022, from https://arxiv.org/abs/1409.1556
39. Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020, April 23). Yolov4: Optimal Speed and accuracy of object detection. arXiv.org. Retrieved July 30, 2022, from https://arxiv.org/abs/2004.10934v1
40. Wikimedia Foundation. (2022, July 27). Residual neural network. Wikipedia. Retrieved July 30, 2022, from https://en.wikipedia.org/wiki/Residual\_neural\_network
41. Ohri, A. (2022, July 6). Rectified Linear Unit (RELU): An important introduction (2021). Jigsaw Academy. Retrieved July 30, 2022, from https://www.jigsawacademy.com/blogs/ai-ml/relu/