# A Study on Convolutional Neural Networks Variant for Face Mask Classification


**Team ID:39** **Presenter: Zihong Sun**

**Team Members: Siriporn Pattamaset, Muhammad Rifki Kurniawan, Casarico Massimo, Qixiu Yang, Galbiati Tiberio, Lu Xu**

# CONTENTS

01

# Introduction

# Introduction

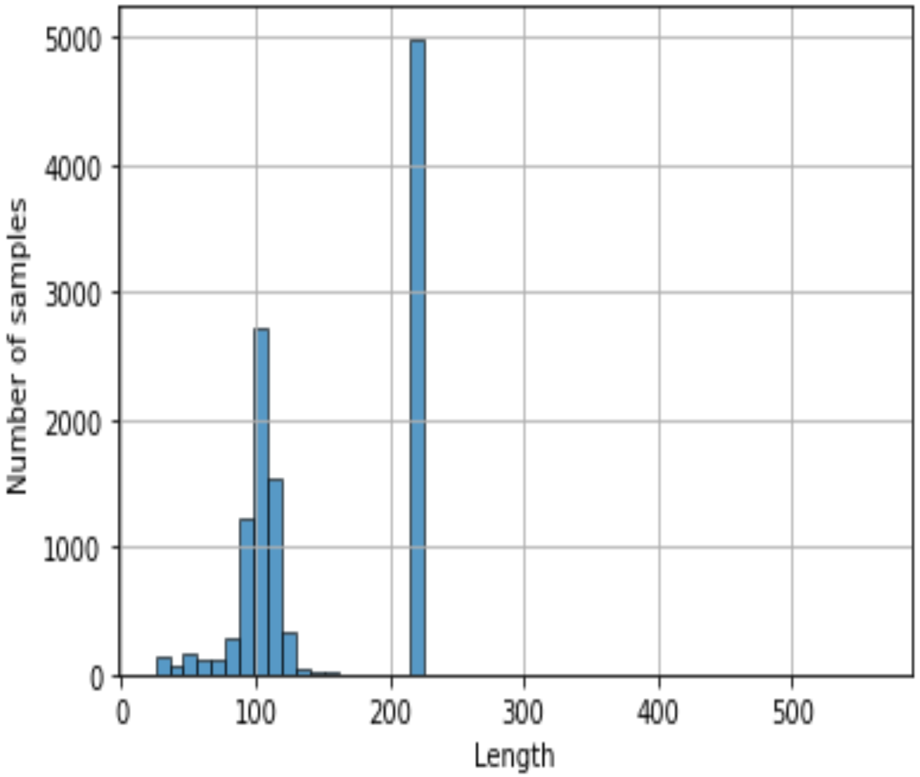
The COVID-19 pandemic has become global pandemic

Testing whether people wear masks has become a new demand

Creating an algorithm that can directly detect is a person is wearing a face mask or not.

# Introduction

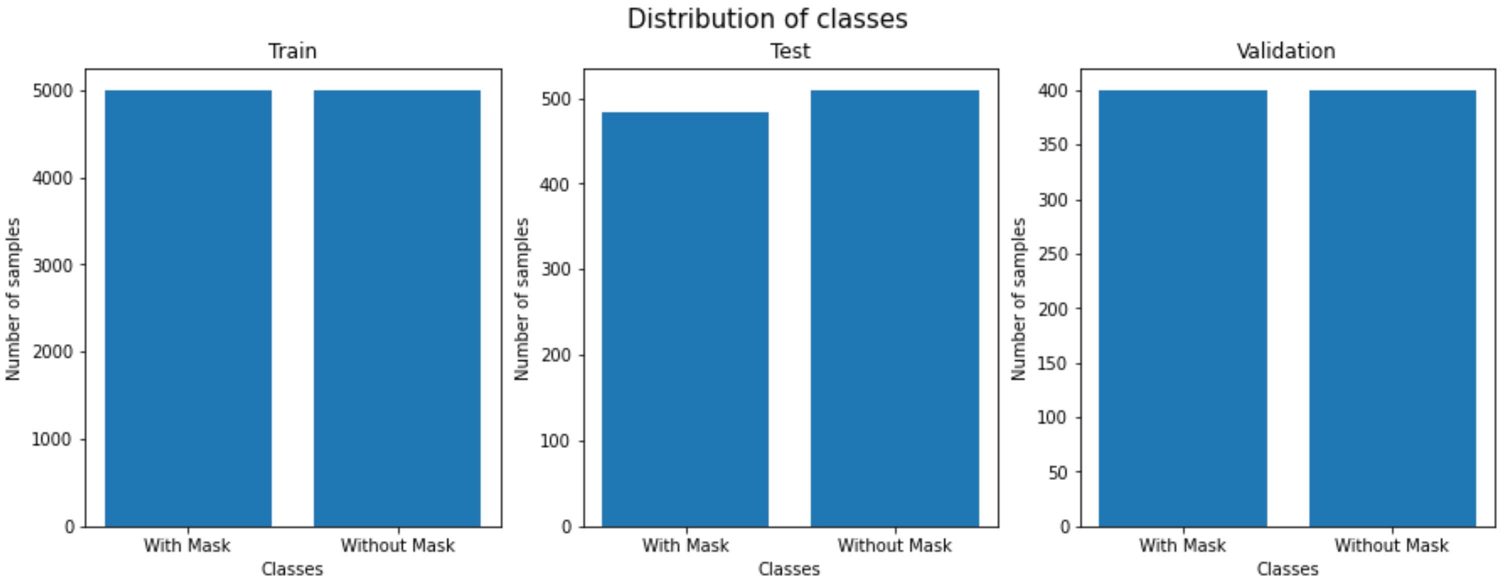
- **Introduction to The Face Mask Data Set**




**Data Set**
12000 images, 328.92MB in size
minimum size: pixels
maximum size: 563 pixels

| Training Set | Test Set | Validation Set |
|---|---|---|
| 10,000 samples | 992 samples | 800 samples |

**The number of the positive images and negative images is balanced in each class**



# Introduction

- **The Face Mask Data examples**

# Introduction

## Deep Learning Frameworks

- TensorFlow
- Keras
- PyTorch

## Binary Classification and Image Classification

- Classification task is two categories
- Distinguish different types of images according to the semantic information of the image

## Transfer learning

The transfer of learned and trained model parameters to a new model to help new model training

# 02 Proposed method

# Proposed method

## Problem Formulation

- 2 classes classification problem formulation

## Transfer Learning

- Formalize transfer learning in term of backbone/headclassifier
- Define loss function (crossentropy/binary-cross-entropy)

## CNN

- feedforward neural network
- including convolution calculations and a deep structure

## Pre-trained ImageNet models

- To solve the problem that the network converges more slowly and the accuracy rate becomes worse.

# Proposed method

**Colaboratory**: A Jupyter notebook environment that requires no setup and runs entirely in the cloud

Creating different models based on the same dataset

Using different architectures and hyperparameters

03

# Experiment and Result

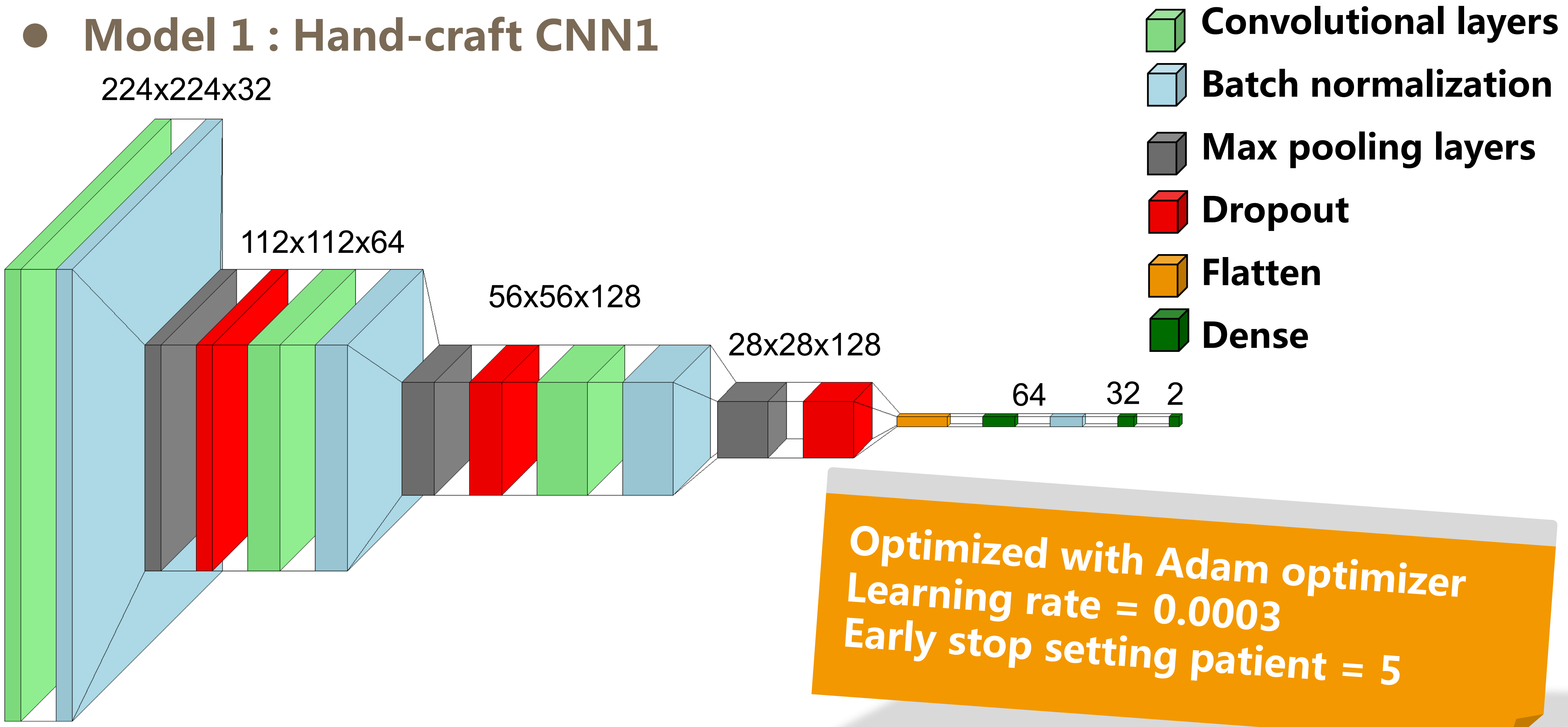
# Experiment and Result

- THE SUMMARY OF PARAMETERS SETUP IN EACH EXPERIMENTS

| | Model | Image shape | Batch size | Learning rate |
|---|---|---|---|---|
| Hand-craft CNN models | Hand-craft CNN1 | (224x224x3) | 32 | 0.0003 |
| | Hand-craft CNN2 | (256x256x3) | 64 | 0.0004 |
| Pre-trained models | EfficientNetB0 | (224x224x3) | 16 | 0.0003 |
| | ResNet18 | (224x224x3) | 32 | 0.001 |
| | Xception | (256x256x3) | 64 | 0.001 |

# Experiment and Result

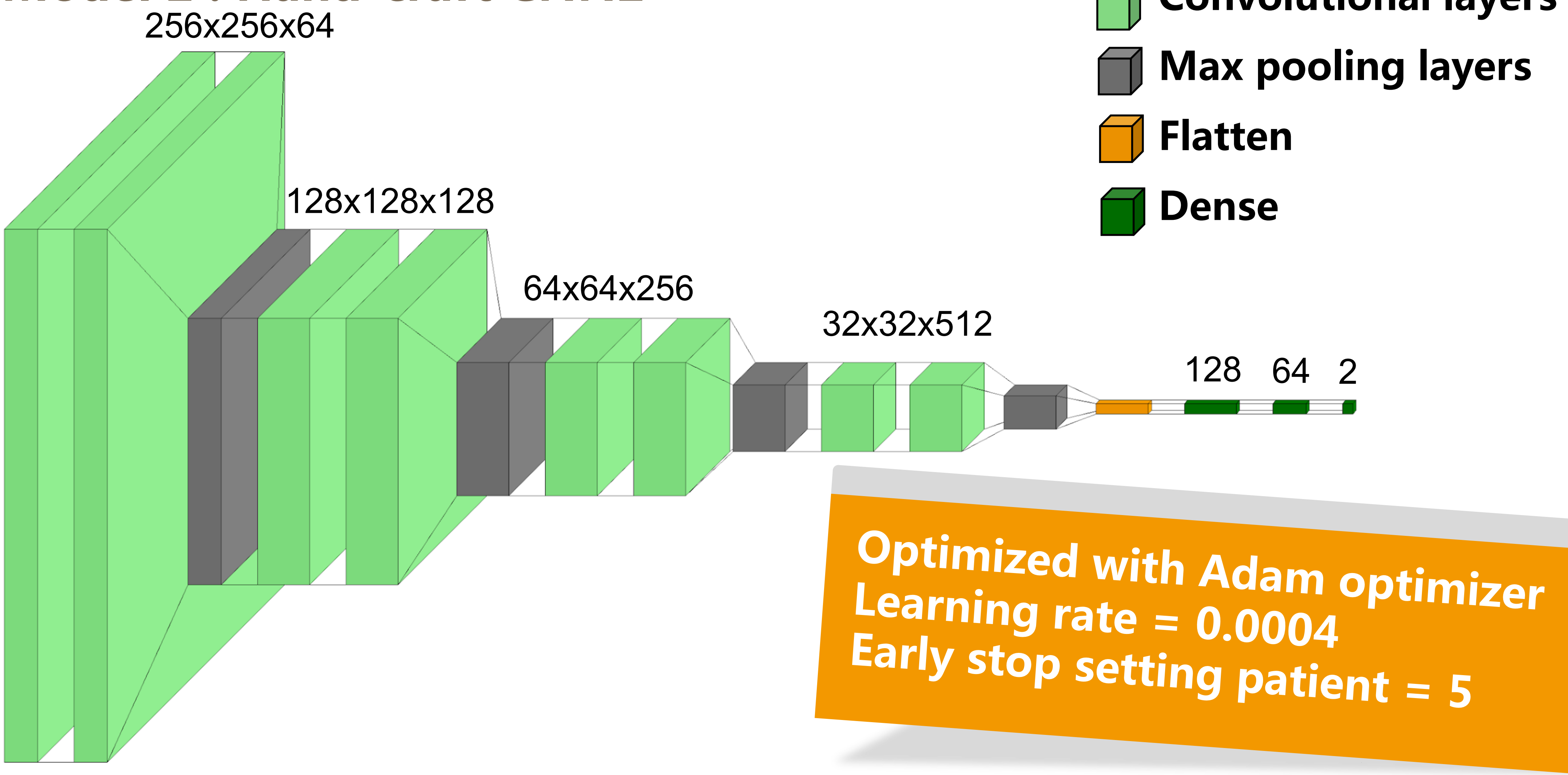
- **Model 1 : Hand-craft CNN1**




**Optimized with Adam optimizer**
**Learning rate = 0.0003**
**Early stop setting patient = 5**

# Experiment and Result

- **Model 2 : Hand-craft CNN2**




Optimized with Adam optimizer
Learning rate = 0.0004
Early stop setting patient = 5

# Experiment and Result

- ## Model 3 : Pre-trained model with EfficientNetB0

| Stage $i$ | Operator $\hat{\mathcal{F}}_i$ | Resolution $\hat{H}_i \times \hat{W}_i$ | #Channels $\hat{C}_i$ | #Layers $\hat{L}_i$ |
|---|---|---|---|---|
| 1 | Conv3x3 | $224 \times 224$ | 32 | 1 |
| 2 | MBConv1, k3x3 | $112 \times 112$ | 16 | 1 |
| 3 | MBConv6, k3x3 | $112 \times 112$ | 24 | 2 |
| 4 | MBConv6, k5x5 | $56 \times 56$ | 40 | 2 |
| 5 | MBConv6, k3x3 | $28 \times 28$ | 80 | 3 |
| 6 | MBConv6, k5x5 | $14 \times 14$ | 112 | 3 |
| 7 | MBConv6, k5x5 | $14 \times 14$ | 192 | 4 |
| 8 | MBConv6, k3x3 | $7 \times 7$ | 320 | 1 |
| 9 | Conv1x1 & Pooling & FC | $7 \times 7$ | 1280 | 1 |

**EfficientNet-B0 baseline network** *(Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." In International Conference on Machine Learning, pp. 6105-6114. PMLR, 2019.)*

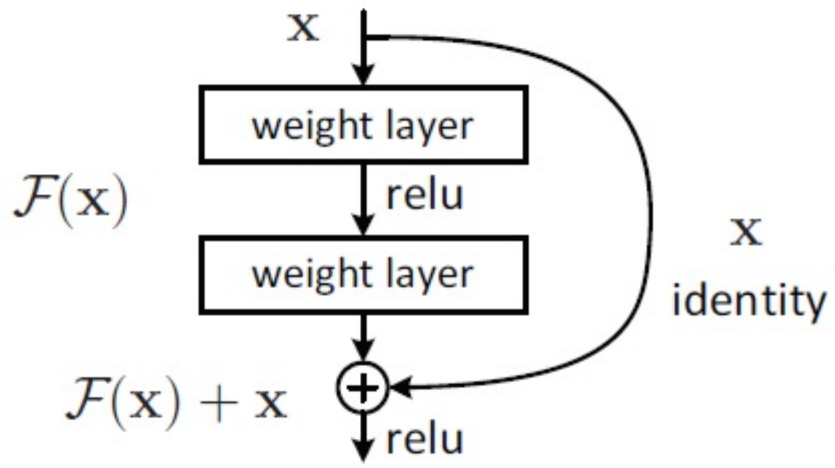
**To apply this pre-trained model to our purpose, we modified classifier from 1,000 classes to 2 classes by attaching fully connected layer**

**Optimized with Adam optimizer**
**weight = pretrained weight**
**Learning rate = 0.00001**
**Early stop setting patient = 5**

# Experiment and Result

- **Model 4 : Pre-trained model with ResNet18**

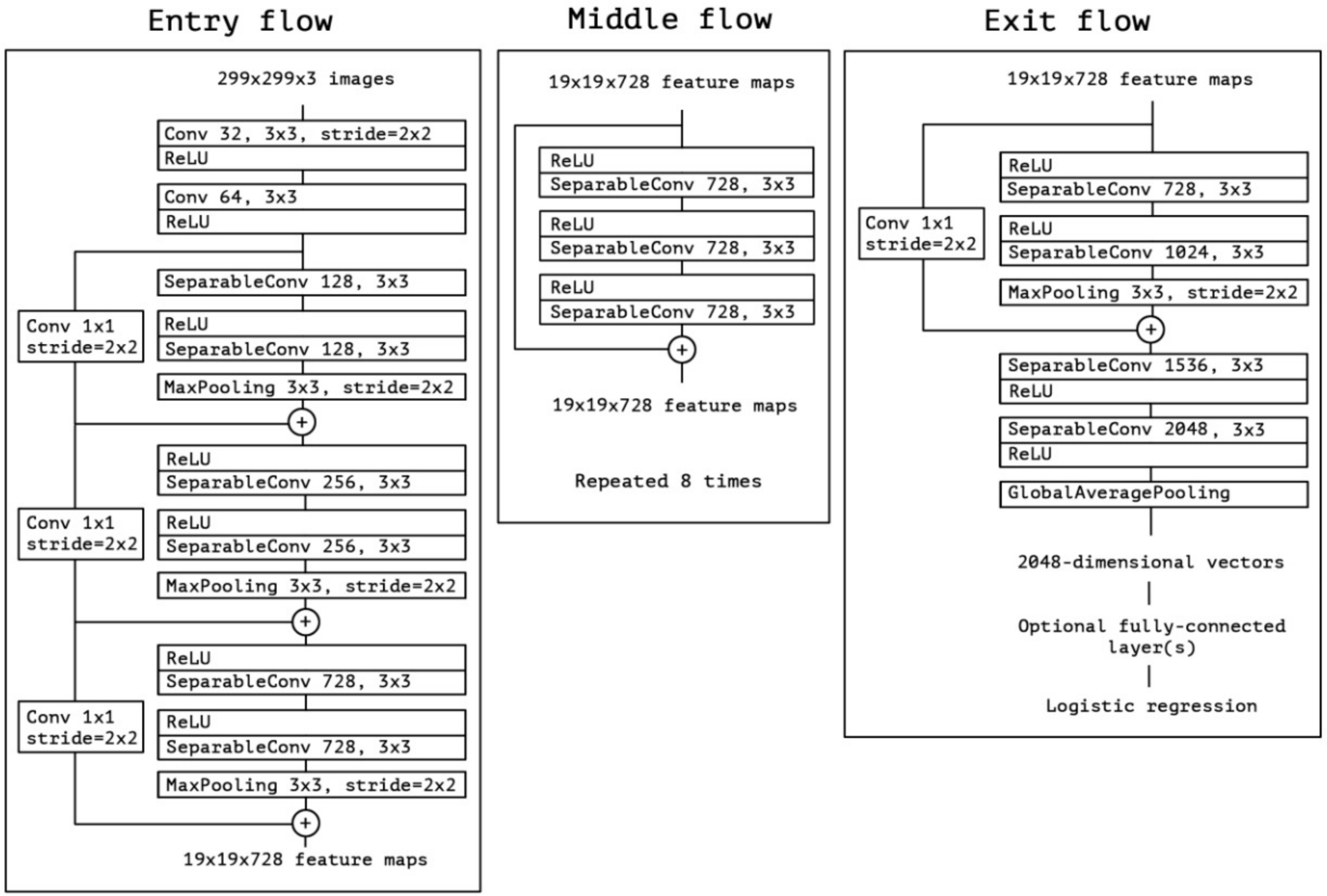
| Layer name | Output size | 18-layer |
|---|---|---|
| Conv1 | 112×112 | 7×7,64,stride 2 |
| Conv2_x | 56×56 | 3×3 max pool, stride 2 |
| | | $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$ |
| Conv3_x | 28×28 | $\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times3$ |
| Conv4_x | 14×14 | $\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times3$ |
| Conv5_x | 7×7 | $\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$ |
| | 7×7 | average pool, 1000-d fc, softmax |




*Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun" Deep Residual Learning for Image Recognition[C] " IEEE Conference on Computer Vision & Pattern Recognition. IEEE Computer Society, 2016.*

# Experiment and Result

## • Model 5 : Pre-trained model with Xception

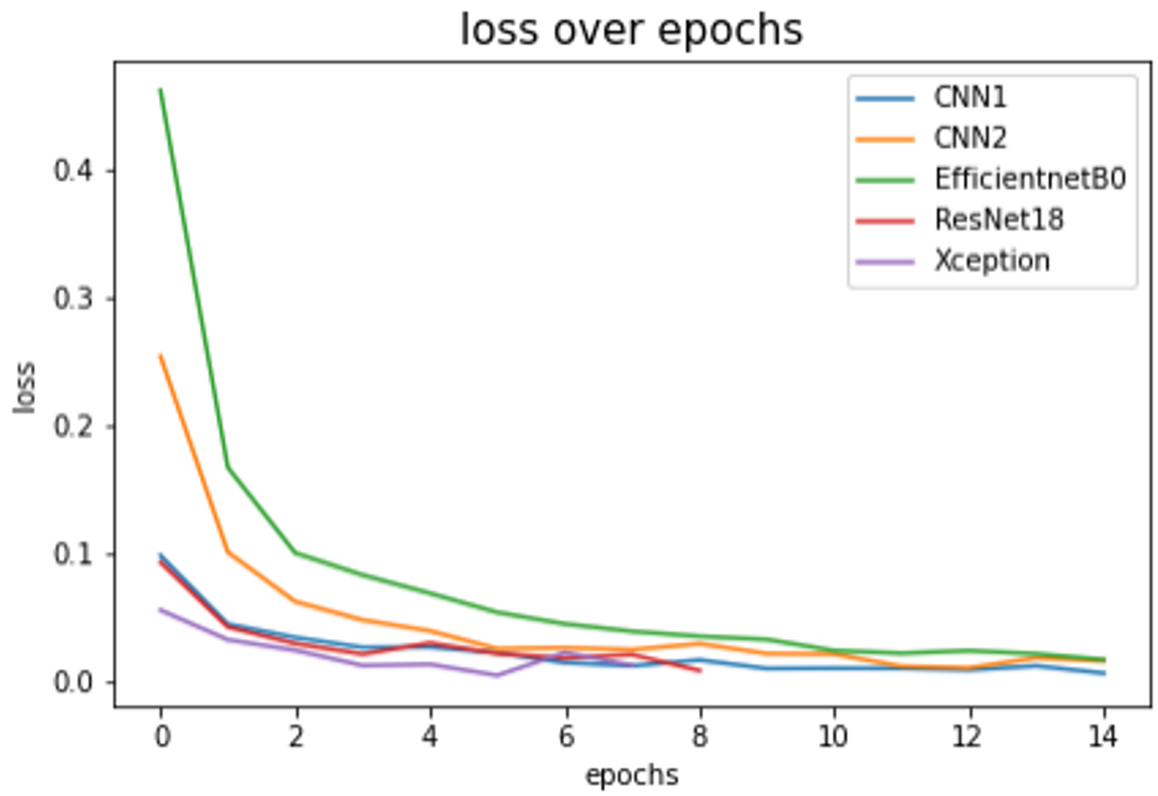



*Chollet, F. . "Xception: Deep Learning with Depthwise Separable Convolutions." 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017).*

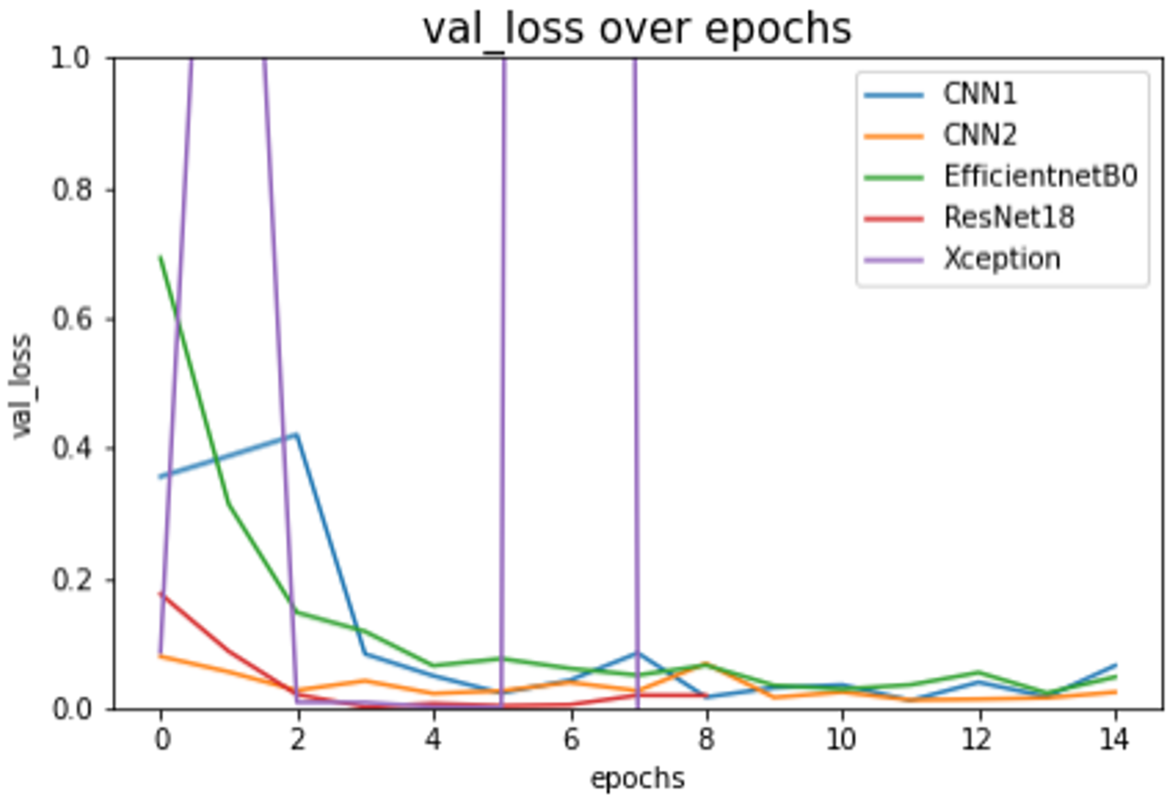
**Tested different version**
- **Cross validation**
- **Using two consecutive trainings**
- **Callback to control the variation of learning rate**

# Experiment and Result

- **Comparing the accuracies in training and validation over epochs**



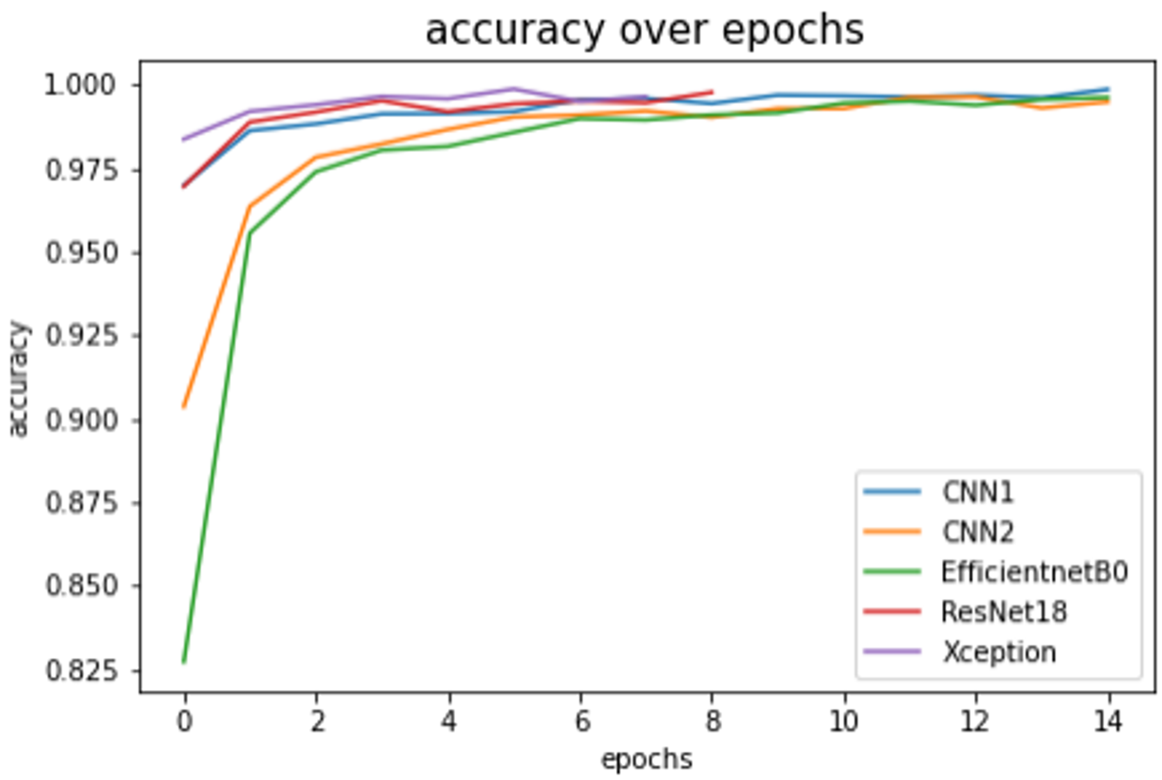

**Training loss**



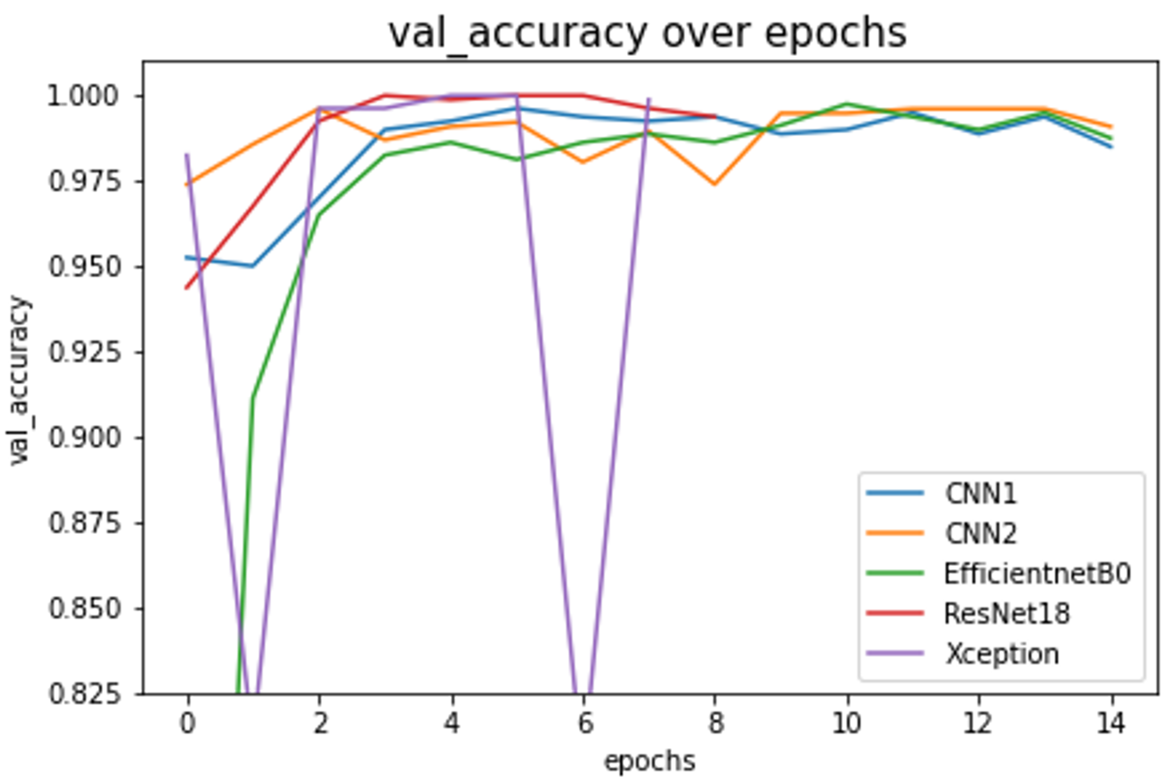

**Validation loss**

# Experiment and Result

- **Comparing the losses in training and validation over epochs**




**Training accuracy**

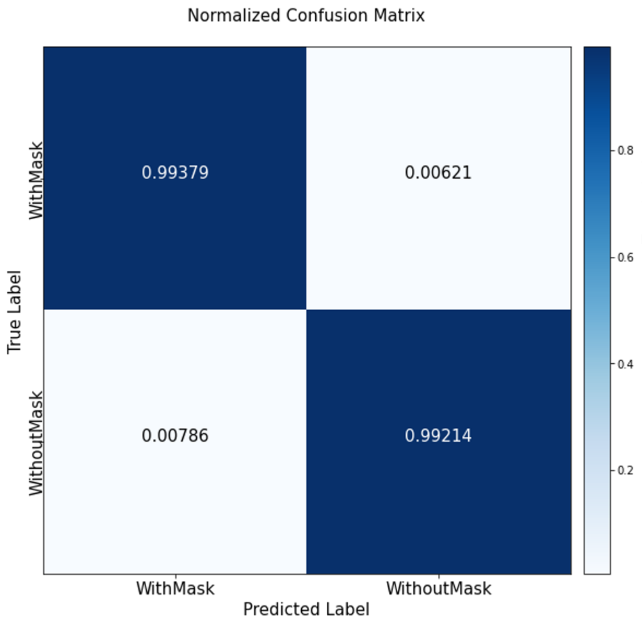


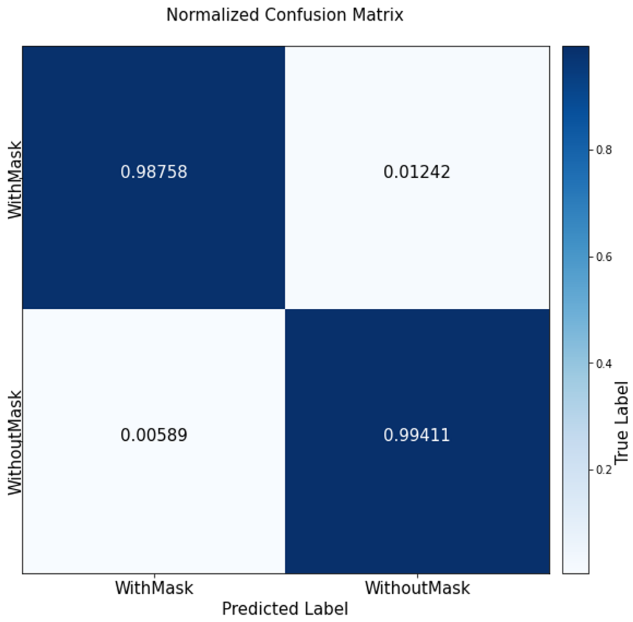

**Validation accuracy**

# Experiment and Result

- **Confusion matrix of the test dataset**

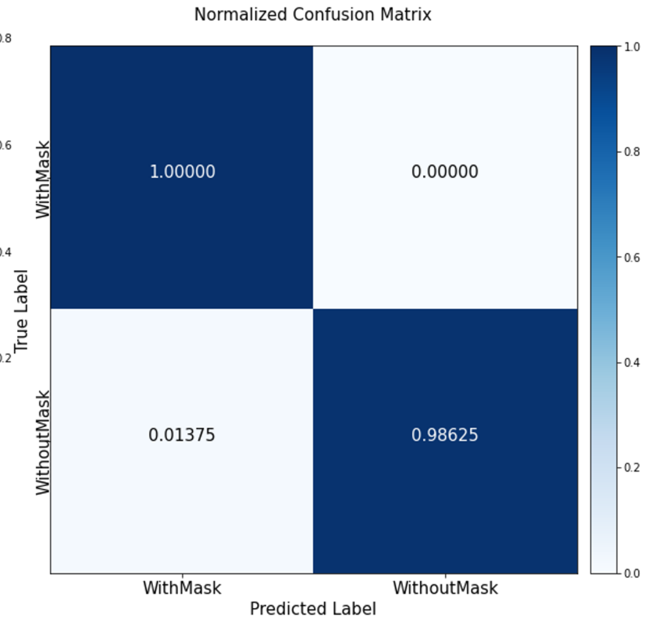
**CNN1**

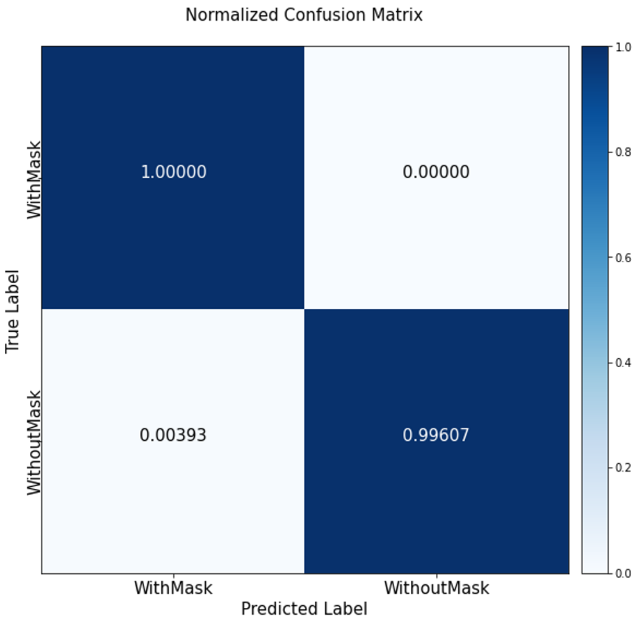



**CNN2**

**EfficientNet0**




**ResNet18**

# Experiment and Result

## ACCURACY AND PERFORMANCE BENCHMARK ON VARIOUS NETWORKS ARCHITECTURES

| Architecture[a] | Weight Initialization[b] | Accuracy (%) | | | | |
|---|---|---|---|---|---|---|
| | | Acc | Precision | Recall | F1-Score | Parameters (Millions) |
| Hand-craft1 | random | 99.3 | 99.3 | 99.3 | 99.3 | 6.5 |
| Hand-craft2 | random | 99.1 | 99.1 | 99.1 | 99.1 | 19.4 |
| ResNet18 [28] | random | 98.9 | 98.8 | 99.2 | 99.0 | 11.2 |
| ResNet18 [28] | pretrained | 99.2 | 98.6 | 100.0 | 99.3 | 11.2 |
| Xception [26] | pretrained | 99.4 | 99.4 | 99.4 | 99.4 | 22.3 |
| MobilenetV2 [29] | random | 99.2 | 98.6 | 100.0 | 99.3 | 2.2 |
| MobilenetV2 [29] | pretrained | 99.7 | 100.0 | 99.6 | 99.8 | 2.2 |
| Efficientnetb0 [25] | random | 99.5 | 99.2 | 100.0 | 99.6 | 4.0 |
| Efficientnetb0 [25] | pretrained | 99.4 | 99.0 | 100.0 | 99.5 | 4.0 |
| MnasNet [30] | random | 98.6 | 98.8 | 98.6 | 98.7 | 3.1 |
| MnasNet [30] | pretrained | 98.6 | 100.0 | 97.4 | 98.7 | 3.1 |

# 04

# Conclusion

# Experiment and Result

## ACCURACY AND PERFORMANCE BENCHMARK ON VARIOUS NETWORKS ARCHITECTURES

| Architecture[a] | Weight Initialization[b] | Accuracy (%) | | | | |
|---|---|---|---|---|---|---|
| | | Acc | Precision | Recall | F1-Score | Parameters (Millions) |
| Hand-craft1 | random | 99.3 | 99.3 | 99.3 | 99.3 | 6.5 |
| Hand-craft2 | random | 99.1 | 99.1 | 99.1 | 99.1 | 19.4 |
| ResNet18 [28] | random | 98.9 | 98.8 | 99.2 | 99.0 | 11.2 |
| ResNet18 [28] | pretrained | 99.2 | 98.6 | 100.0 | 99.3 | 11.2 |
| Xception [26] | pretrained | 99.4 | 99.4 | 99.4 | 99.4 | 22.3 |
| MobilenetV2 [29] | random | 99.2 | 98.6 | 100.0 | 99.3 | 2.2 |
| MobilenetV2 [29] | pretrained | 99.7 | 100.0 | 99.6 | 99.8 | 2.2 |
| Efficientnetb0 [25] | random | 99.5 | 99.2 | 100.0 | 99.6 | 4.0 |
| Efficientnetb0 [25] | pretrained | 99.4 | 99.0 | 100.0 | 99.5 | 4.0 |
| MnasNet [30] | random | 98.6 | 98.8 | 98.6 | 98.7 | 3.1 |
| MnasNet [30] | pretrained | 98.6 | 100.0 | 97.4 | 98.7 | 3.1 |

# Conclusion

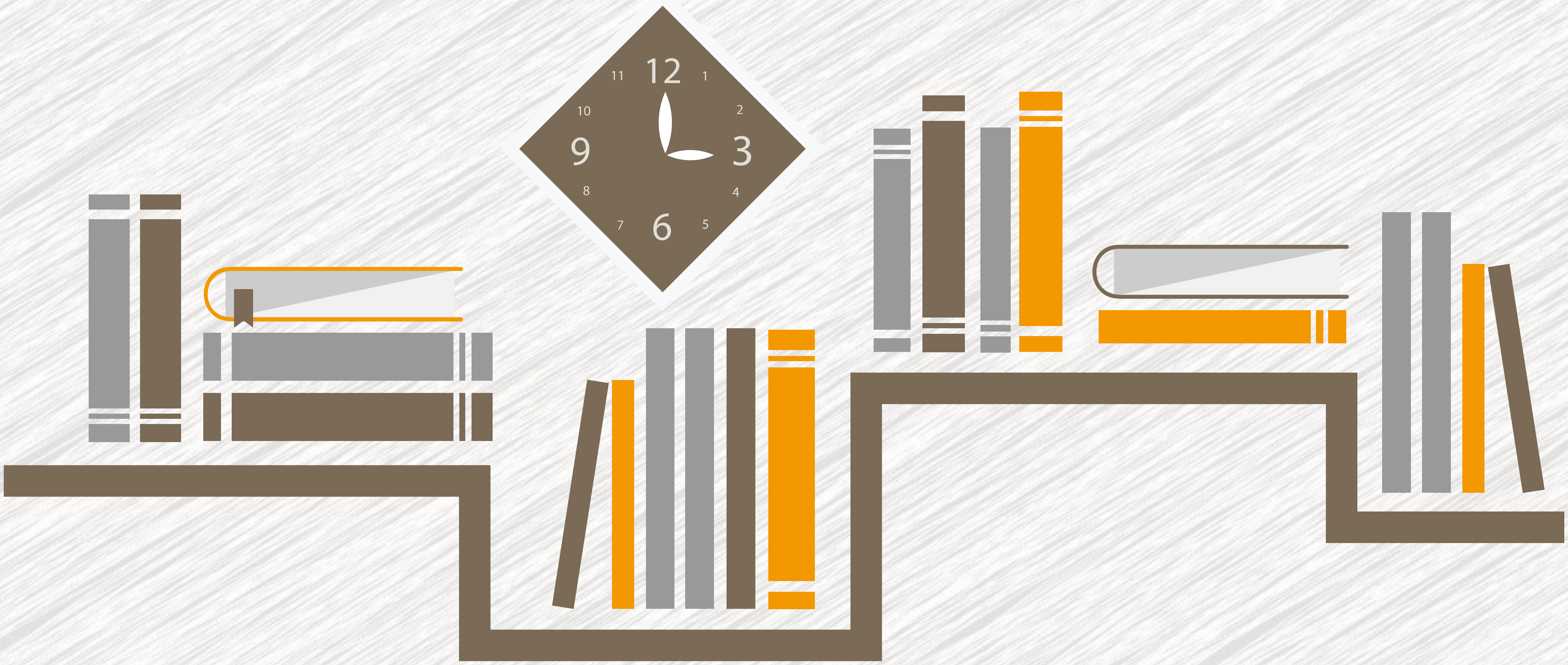
- All the experimental models can achieve the accuracy over 99%.
- The dataset seems well prepared, so all the models performed accurately and achieve over 99\% accuracy.
- MobilenotV2 can give the best performance over the other models.

## Models with Transfer learning

- Training the model with pre-trained weights initialization can yield better than random weights initialization.
- Compared to other pre-trained models achieving similar accuracy, MobilenetV2 is much smaller than MnasNet, EfficientNetB0, and others.

## Simple CNNs models

- the simple model of CNNs, such as hand-craft CNN1, provides simple architecture, less complexity with high accuracy as well.



# THANK YOU