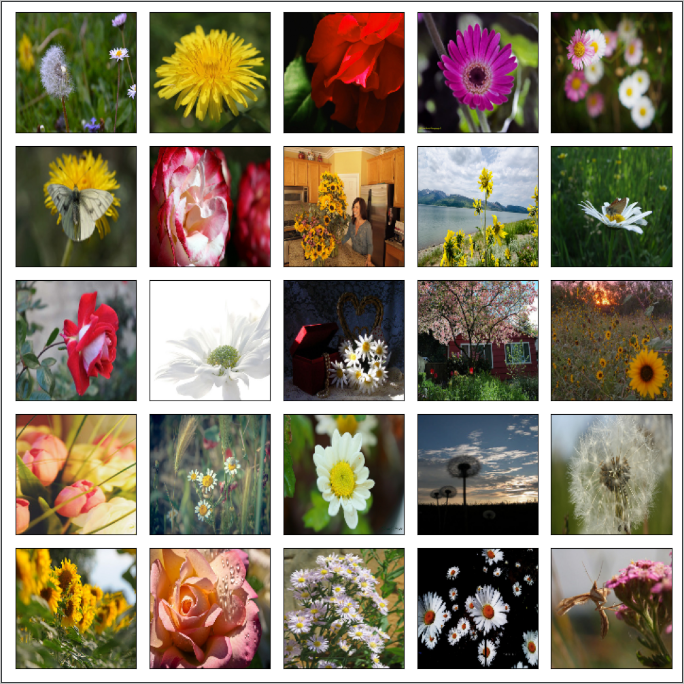


**Data Augmentation and Neural Networks**

**To**

**Improve Image Classification**

FINAL PROJECT: MACHINE LEARNING II



Joseph Francis

Srilatha Lakka

Professor Amir Jafari

DATS6203

Table of Contents

[**1.** **INTRODUCTION** 4](#_Toc70099393)

[1.1 Problem Statement 4](#_Toc70099394)

[1.2 Project Outline 4](#_Toc70099395)

[**2.** **DATA SET** 5](#_Toc70099396)

[2.1 Resource 5](#_Toc70099397)

[2.2 Overview 5](#_Toc70099398)

[**3.** **NEURAL NETWORK ARCHITECTURE** 6](#_Toc70099399)

[**3.1** **INCEPTIONV3** 6](#_Toc70099400)

[3.1.1 Description 6](#_Toc70099401)

[3.1.2 Configuration 6](#_Toc70099402)

[3.1.3 Architecture 6](#_Toc70099403)

[**3.2** **XCEPTION** 6](#_Toc70099404)

[3.2.1 Description 6](#_Toc70099405)

[3.2.2 Configuration 6](#_Toc70099406)

[3.2.3 Architecture 6](#_Toc70099407)

[**3.3** **RESNET50** 6](#_Toc70099408)

[3.3.1 Description 6](#_Toc70099409)

[3.3.2 Configuration 6](#_Toc70099410)

[3.3.3 Architecture 6](#_Toc70099411)

[**4.** **FCGAN** 7](#_Toc70099412)

[**5.** **EXPERIMENTAL SETUP** 7](#_Toc70099413)

[5.1 Pre-Processing 7](#_Toc70099414)

[5.1.1. Train, Validation Test Split 7](#_Toc70099415)

[5.2 No Augmentation 8](#_Toc70099416)

[5.3 Conventional Data Augmentation 9](#_Toc70099417)

[5.4 FC GAN 9](#_Toc70099418)

[**6.** **RESULTS** 9](#_Toc70099419)

[6.1 Accuracy Report 9](#_Toc70099420)

[6.2 Performance Metrics 9](#_Toc70099421)

[6.3 FCGAN 9](#_Toc70099422)

[**7.** **KEY INSIGHTS** 9](#_Toc70099423)

[**8.** **CONCLUSION** 9](#_Toc70099424)

[**9.** **FURTUE PROSPECTS** 9](#_Toc70099425)

[**10.** **REFERENCES** 10](#_Toc70099426)

[**11.** **APPENDIX** 12](#_Toc70099427)

# **INTRODUCTION**

In machine learning, image classification is a process to analyze the extracted image features and organize them into categories by using neural networks. In recent years, the neural network techniques have improved image classification accuracy quickly, such as AlexNet can achieve 15.3% image classification error rates1 . These techniques are practically applied in many fields, such as artificial intelligence, medical diagnostics, E-commerce, gaming, or automotive industries. The convolutional neural network (CNN) is one of the most popular deep neural network (DNN) learning algorithms which can perform classification tasks directly from images. CNN models can produce the state-of-the-art classification predictions with directly learned features. A CNN model generally contains more than one convolutional layer for specialized linear operations and includes local or global pooling layers to perform nonlinear down sampling. After learning features from many layers, a fully connected layer outputs the probabilities for each class to be predicted. However, model overfitting and poor performance are common problems in applying neural network techniques.

## Problem Statement

Model overfitting and poor performance are common problems in applying neural network techniques because some of the high frequency features may not be useful in classification . Approaches to bring intra-class differences down and retain sensitivity to the inter-class variations are important to maximize model accuracy and minimize the loss function.

## 1.2 Project Outline

Data augmentation is a common technique to overcome the lack of large, annotated databases, a usual situation when applying deep learning to medical imaging problems. Nevertheless, there is no consensus on which transformations to apply for a particular field. This project aims at identifying the effect of different transformations on images using deep learning and various pretrained models such as InceptionV3, Xception and ResNet50 and finally use Generative Adversarial Networks or GAN.

We will first introduce the data, then concepts, algorithms, structures of our various models and their theories. Then we'll explain each step of implementation, the challenges we encountered, and how we arrived in solving. Finally, we will explain the conclusion and the ideas for future improvement.

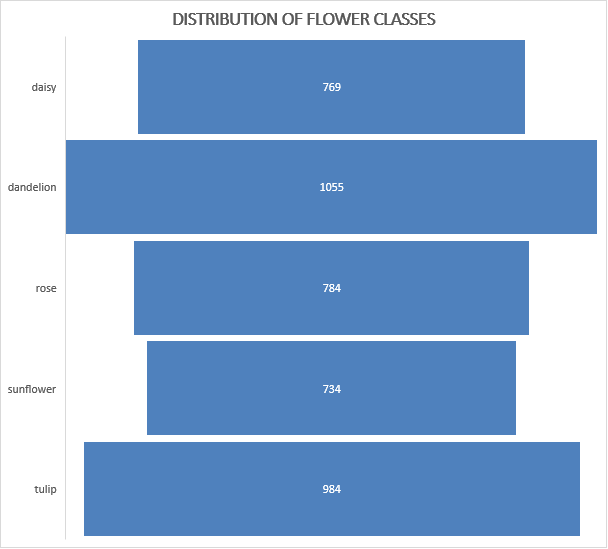
# **DATA SET**

## Resource

This dataset contains 4242 images of flowers. The data collection is based on the data flicr, google images, and yandex images. This can be used to recognize plants from the photo.

## Overview

The pictures are divided into five classes: daisy, tulip, rose, sunflower, and dandelion. For each class there are about 800 photos. Photos are not high resolution, about 320x240 pixels. Photos are not reduced to a single size; they have different proportions!





j

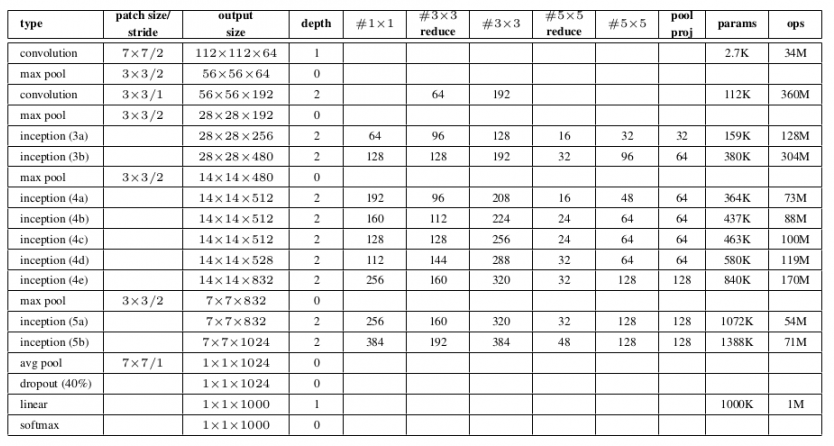
# **NEURAL NETWORK ARCHITECTURE**

## 3.1. INCEPTION v3

## Description

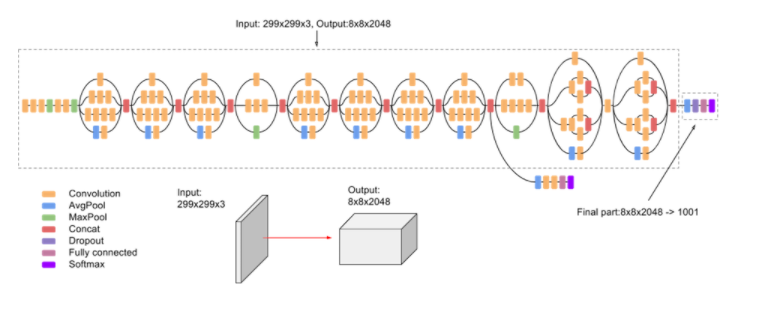
**Inception v3** is a convolutional neural network for assisting in image analysis and object detection, and got its start as a module for Googlenet. It is the third edition of Google’s inception Convolutional Neural Network, originally introduced during the ImageNet Recognition Challenge.

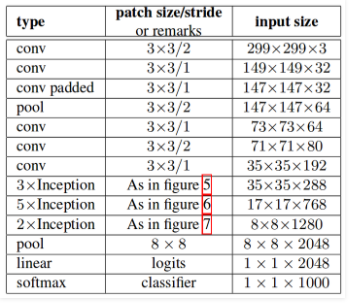
## Configuration



*Fig 1: Source: https://cdn.analyticsvidhya.com/wp-content/uploads/2018/10/Screenshot-from-2018-10-16-11-56-41-850x462.png*

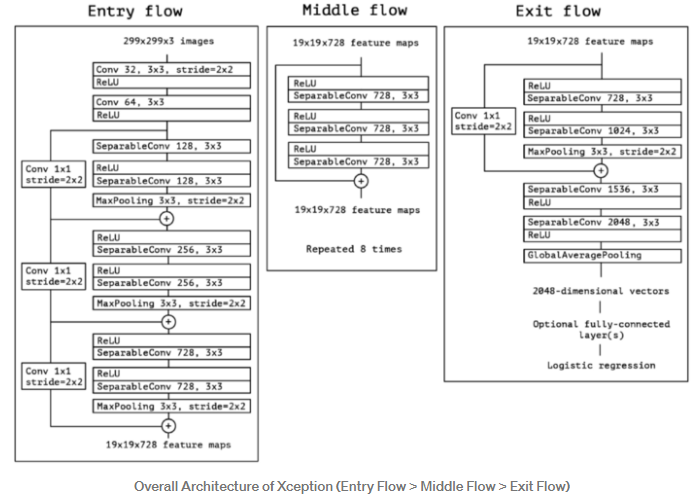
## Architecture

 Inception-v3is a convolutional neural network architecture from the Inception family that makes several improvements including using   
Label Smoothing, Factorized 7 x 7   
convolutions, and the use of an auxiliary classifier to propagate label information lower down the network (along with the use of batch normalization for layers in the sidehead)

  
 *Fig 2: Source:* [*https://www.geeksforgeeks.org/inception-v2-and-v3-inception-network-versions/*](https://www.geeksforgeeks.org/inception-v2-and-v3-inception-network-versions/)

## 3.2.1 XCEPTION

Xception is an extension of the inception Architecture which replaces the standard Inception modules with depthwise Separable Convolutions.

As in the figure above, SeparableConv is the modified depthwise separable convolution. We can see that SeparableConvs are treated as Inception Modules and placed throughout.

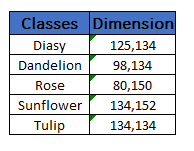
## 3.3.1 RESNET50

ResNet is a powerful backbone model that is used very frequently in many computer vision tasks. ResNet50 is a variant of ResNet model which has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer. It has 3.8 x 10^9 Floating points operations.

## *Fig 3: Source:* [*https://www.geeksforgeeks.org/inception-v2-and-v3-inception-network-versions/*](https://www.geeksforgeeks.org/inception-v2-and-v3-inception-network-versions/)

# **EXPERIMENTAL SETUP**

## Pre-Processing

The image dimensions are not uniform, after loading them, they were resized before split to training part and validation part. In this place, we tried **150 x 150.**

## 5.1.1. Train, Validation Test Split



For training and testing purposes for our model, we have our data broken down into three distinct datasets. These datasets will consist of the following:

**Training set:** It’s the set of data used to train the model. During each epoch, our model will be trained over and over again on this same data in our training set, and it will continue to learn about the features of this data. Later we can deploy our model, and have it accurately predicted on new data that it’s never seen before. It will be making these predictions based on what it’s learned about the training data.

**Validation set:** This is a set of data, separate from the training set that is used to validate our model during training. This validation process helps give information that may assist us with adjusting our hyper parameters. With each epoch during training, the model will be trained on the data in the training set and it will also simultaneously be validated on the data in the validation set. During the training process, the model will be classifying the output for each input in the training set. After this classification occurs, the loss will then be calculated, and the weights in the model will be adjusted. Then, during the next epoch, it will classify the same input again.

Note: The data in the validation set is separate from the data in the training set. When the model is validating on this data, this data does not consist of samples that the model already is familiar with from training to ensure that our model is not overfitting to the data in the training set.

**Test set**: This is a set of data that is used to test the model after the model has already been trained. The test set is separate from both the training set and validation set. After our model has been trained and validated using our training and validation sets, we will then use our model to predict the output of the data in the test set.

## No Augmentation

two convolutional layers, one max-pooling layer, and one classifier layer for image classification

We start with the basic CNN model with two use convolutional layers because they are spatially invariant and great for images. This means that no matter where an element appears in an image it can be detected. This is because conv layers use filters which slide over the image/volumes to produce activation maps.

The max pooling layers generally come after the convolutional layers. They shrink the size of the volume by taking the max value within the filter and hence reduce computation costs as we move through the network.

ReLU's or Rectified Linear Units are used as the activation function because they help reduce vanishing gradients which are an issue with sigmoid and TanH activation layers. Vanishing gradients are a problem because they reduce the likelihood that a signal will propagate to the input layer and adjust those weights. Hence the earlier layers will not learn

The last layer has a softmax activation with 5 units to account for the 5 classes of the dataset. We will get a probability for each of the classes.

We used Adam optimizer because it adjusts the learning rate separately per layer and can be used with weight decay so that when we get close to convergence the learning can be reduced so we can thus get better results. Categorical crossentropy is chosen since this is a multiclass classification problem.

We use the fit\_generator here because we are getting the data from an ImageDataGenerator. These correspond to generators in python and can produce batches into infinity. These correspond to generators in python and can produce batches into infinity. Running for 30 epochs we see that the best validation accuracy is about **0.6440**, We still have some work to do!

## Conventional Data Augmentation

Data augmentation techniques are often used to regularize models which work with images in neural networks and other learning algorithms. With the labelled original training dataset, synthetic images can be created by various transformations to the original images.

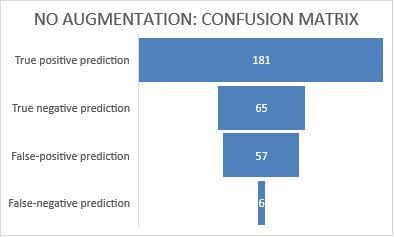
Keras ImageDataGenerators11 is the tool used for generating more training data from the original data to avoid model overfitting. It is conducted online by looping over in small batches during each optimizer iteration. There are some graphic parameters (e.g. rotation, shift, flip, add Gaussian noises) to help generate artificial images. In Figure 6, some example images generated by using the ImageDataGenerator are shown. Importantly, these images are not included in the original CIFAR-10 image dataset. These generated new image data are mini-batched and discarded after model training.

There are various data augmentation techniques: (1) flipping images horizontally or vertically; (2) rotating images at some degrees; (3) rescaling outward or inward; (4) randomly cropping; (5) translating by width and height shifts; (6) whitening, (7) shearing, (8) zooming and (9) adding 10% Gaussian noises to prevent model over-fitting and enhance learning capability

# **RESULTS**

## Accuracy Report

## Performance Metrics

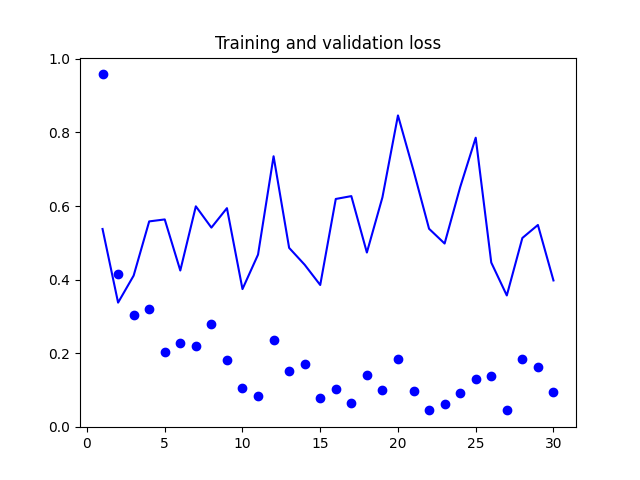
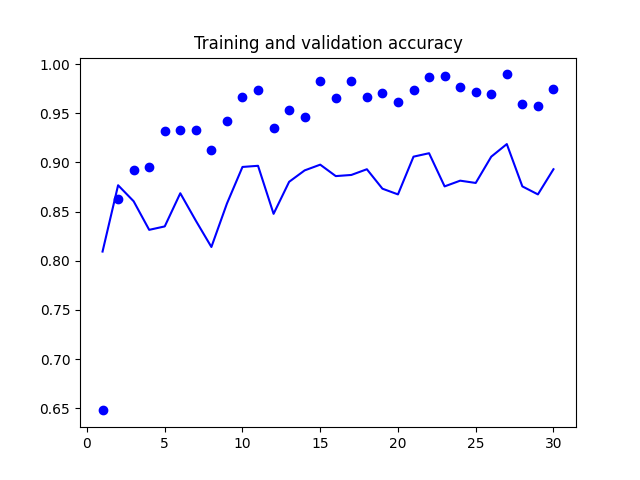


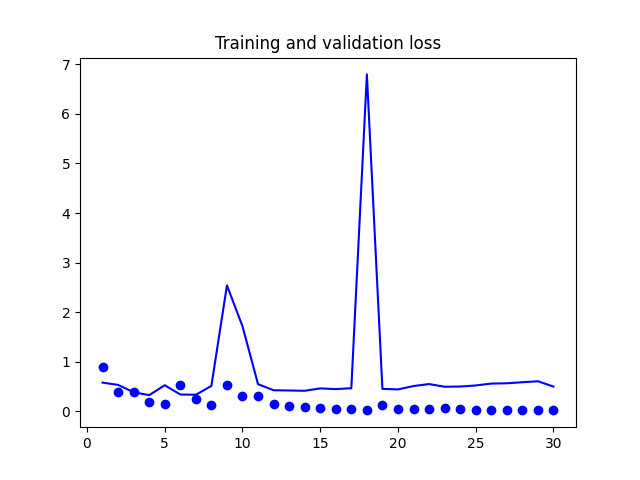
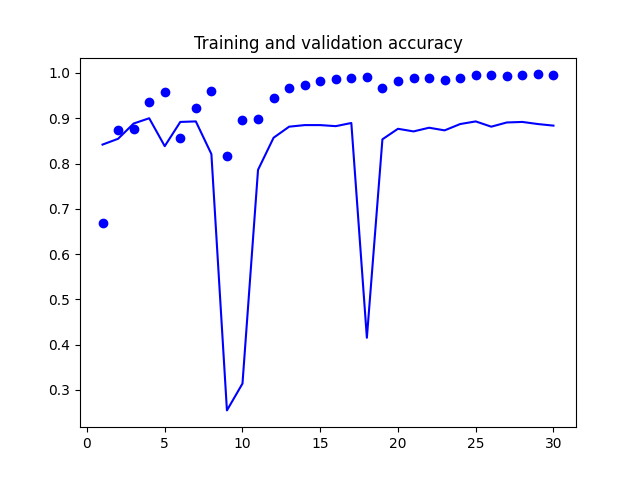
## *Fig 4: No Augmentation: Metrics Fig 5: Confusion Matrix*

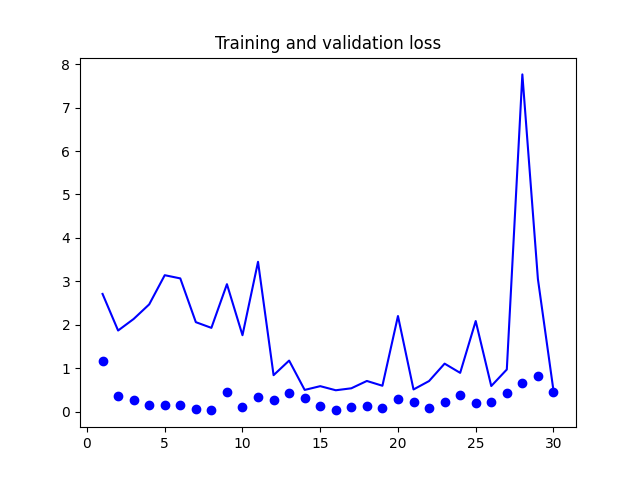
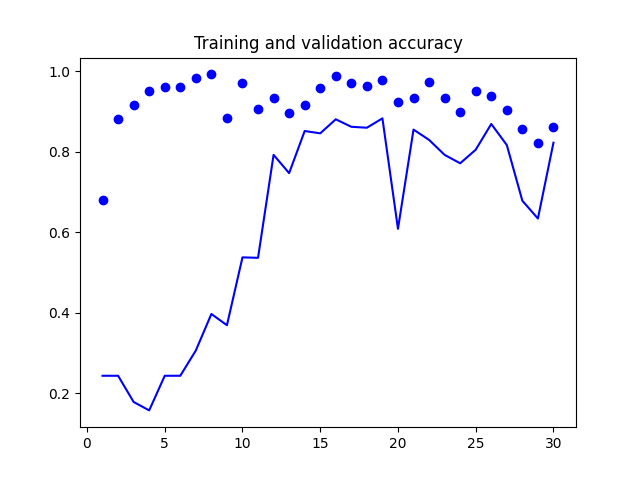
## 

## *Fig 6: Data Augmentation: Metrics Fig 7: Data Aug. Confusion Matrix*

## 5.3. Training & Validation Loss (no augmentation)



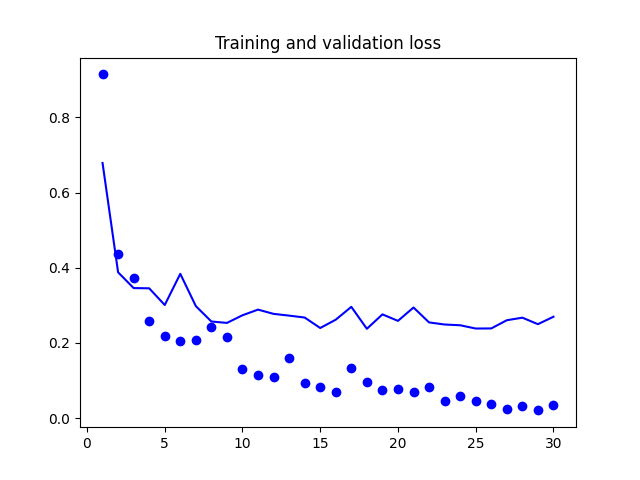
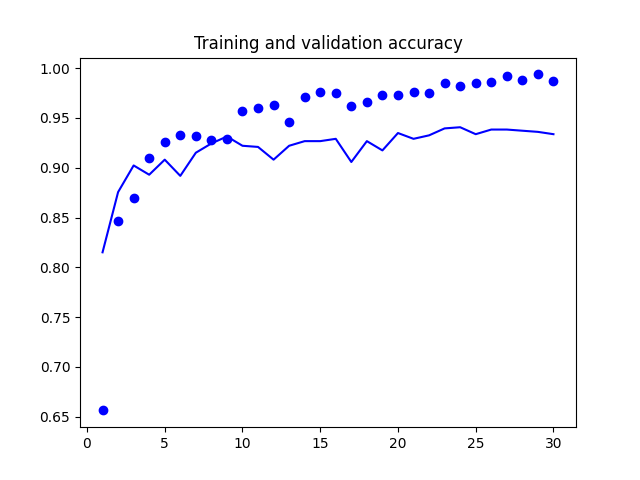
*Fig 8: Inception accuracy Fig 9: Inception loss*  
 *Fig 10: Xception accuracy Fig 11: Xception loss*



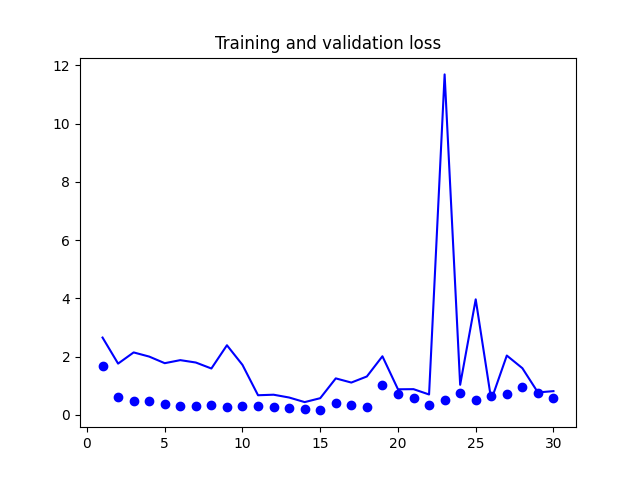
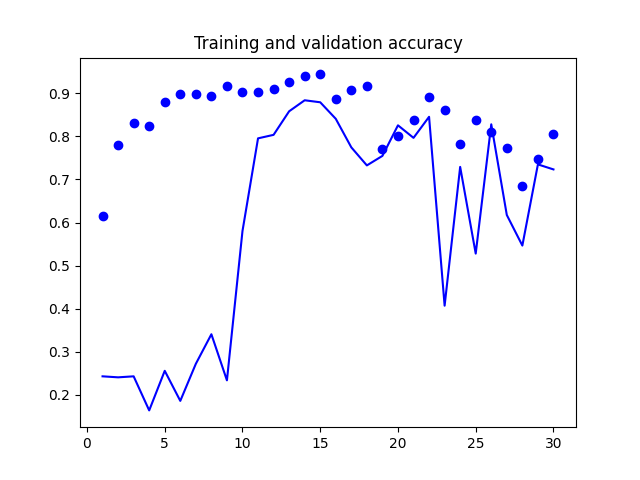
*Fig 12: Resnet50 accuracy Fig 13: Resnet50 loss*

## 5.4. Training & Validation Loss (with augmentation) inception_train&val_accuracyinception_train&val_loss

*Fig 14: Inception accuracy Fig 15: Inception loss*



*Fig 16: Xception accuracy Fig 17: Xception loss*

**

*Fig 18: Resnet50 accuracy Fig 19: Resnet50 loss*

# 

# **KEY INSIGHTS**

# **CONCLUSION**

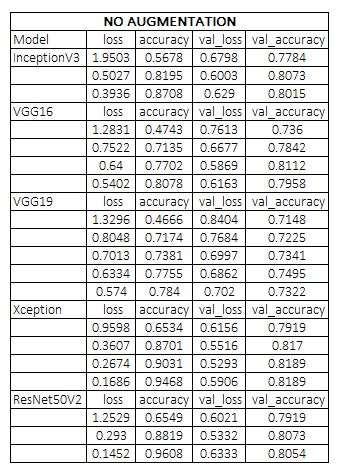
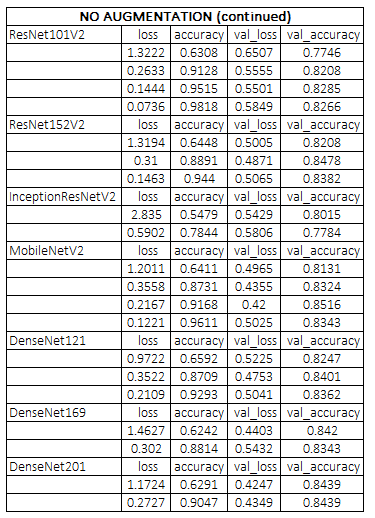
# **FURTUE PROSPECTS**

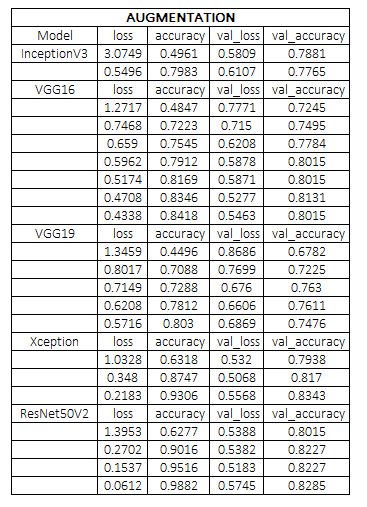
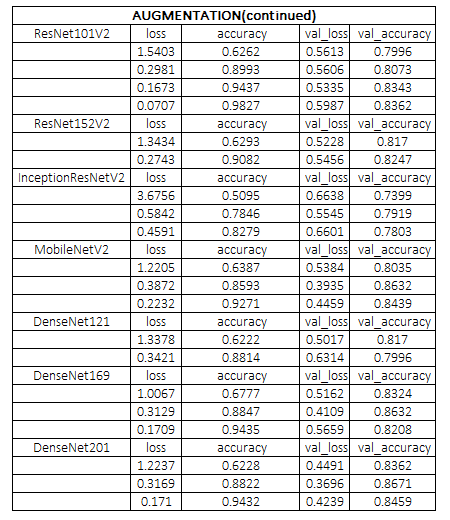
# **REFERENCES**

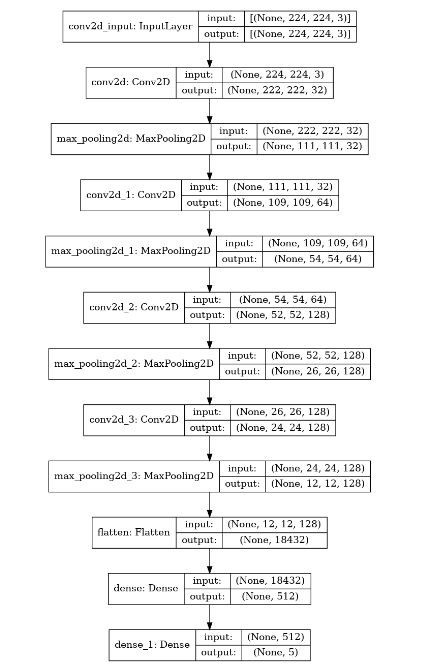
[1] keras.io

[2] Pratt, L. Y. (1993). [“Discriminability-based transfer between neural networks”](http://papers.nips.cc/paper/641-discriminability-based-transfer-between-neural-networks.pdf) (PDF). NIPS Conference: Advances in Neural Information Processing Systems 5. Morgan Kaufmann Publishers. pp. 204–211.  
  
[3] Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. IJCV, 2015.  
  
[4] [LeCun, Y. et al., 1998. “Gradient-based learning applied to document recognition.” Proceedings of the IEEE, 86(11):2278–2324](http://yann.lecun.com/exdb/publis/index.html#lecun-98)  
  
[5] Han X., Kashif R., and Roland V., 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms [arXiv preprint](https://arxiv.org/abs/1708.07747)  
  
[6] Chollet F., 2016. Xception: Deep Learning with Depthwise Separable Convolutions [arXiv preprint](https://arxiv.org/abs/1610.02357)  
  
[7] Simonyan K. and Zisserman A., 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition [arXiv preprint](https://arxiv.org/abs/1409.1556)  
  
[8] Szegedy C. et al., 2016. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning [arXiv preprint](https://arxiv.org/abs/1602.07261)  
  
[9] Sandler M. et al., 2019. MobileNetV2: Inverted Residuals and Linear Bottlenecks [arXiv preprint](https://arxiv.org/pdf/1801.04381.pdf)  
  
[10] Zoph B. et al., 2018. Learning Transferable Architectures for Scalable Image Recognition [arXiv preprint](https://arxiv.org/pdf/1707.07012.pdf)  
  
[11] He K. et al., 2016. Deep Residual Learning for Image Recognition [arXiv preprint](https://arxiv.org/abs/1512.03385)  
  
[12] Huang G. et al., 2017. Densely Connected Convolutional Networks [arXiv preprint](https://arxiv.org/abs/1608.06993)  
  
[13] Szegedy C. et al., 2016. Rethinking the Inception Architecture for Computer Vision [arXiv preprint](https://arxiv.org/abs/1512.00567)

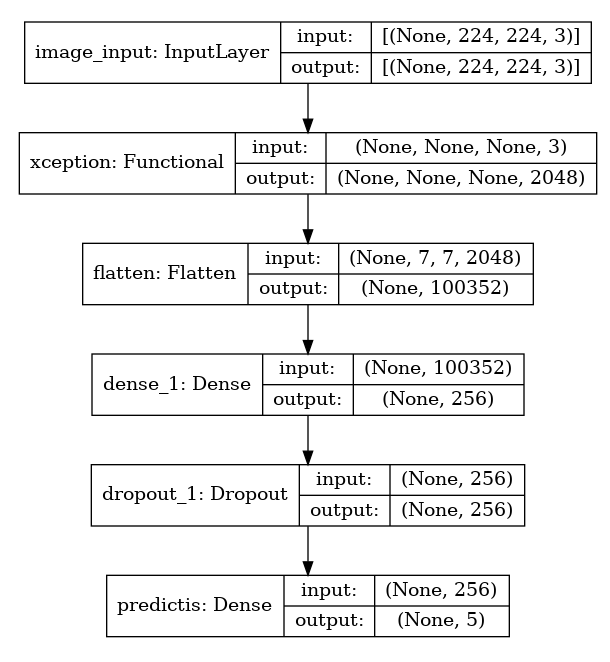
# **APPENDIX**



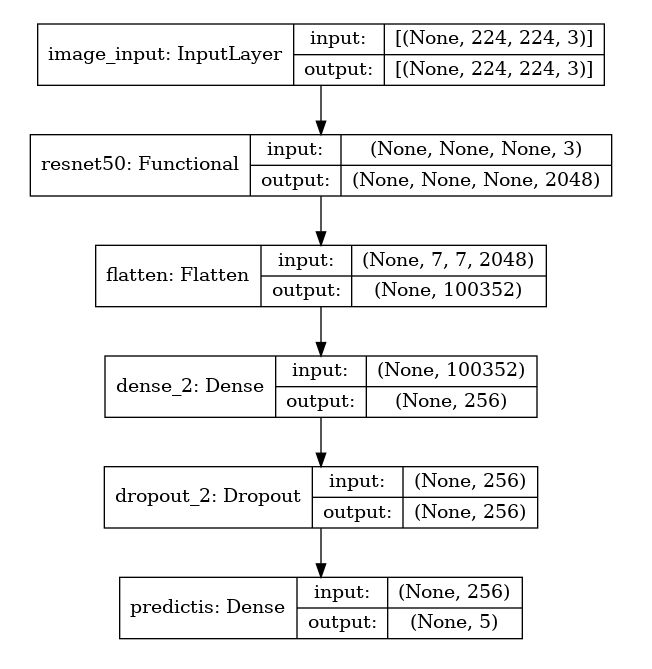




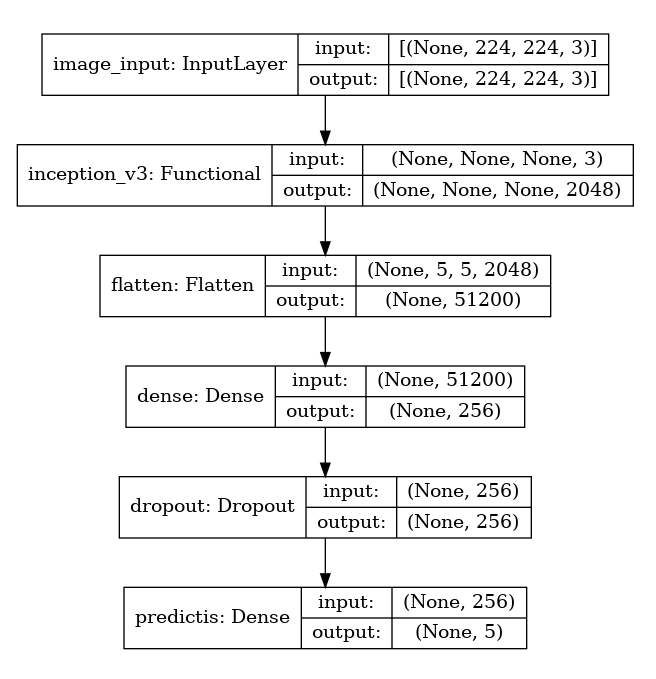
**MODEL SUMMARY PLOTS**

***Model1: Baseline***

***Model: Xception***



***Model2*:*In***

***Model4: Resnet50***