

# Food Image Recognition and Calorie Estimation Using Object Detection Algorithms

MSc Research Project  
Data Analytics

Manoj Kumar Yuganathan  
Student ID:x20179189

School of Computing  
National College of Ireland

Supervisor: Professor Majid Latifi

National College of Ireland  
Project Submission Sheet  
School of Computing



<b>Student Name:</b>	Manoj Kumar Yuganathan
<b>Student ID:</b>	x20179189
<b>Programme:</b>	Data Analytics
<b>Year:</b>	2021-2022
<b>Module:</b>	MSc Research Project
<b>Supervisor:</b>	Professor Majid Latifi
<b>Submission Due Date:</b>	31/01/2022
<b>Project Title:</b>	Food Image Recognition and Calorie Estimation Using Object Detection Algorithms
<b>Word Count:</b>	1400
<b>Page Count:</b>	9

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

**ALL** internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

<b>Signature:</b>	Manoj Kumar Yuganathan
<b>Date:</b>	31st January 2022

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:**

Attach a completed copy of this sheet to each project (including multiple copies).	<input type="checkbox"/>
<b>Attach a Moodle submission receipt of the online project submission</b> , to each project (including multiple copies).	<input type="checkbox"/>
<b>You must ensure that you retain a HARD COPY of the project</b> , both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

# Food Image Recognition and Calorie Estimation Using Object Detection Algorithms

Manoj Kumar Yuganathan  
x20179189

## 1 Introduction

In the Food-101 dataset <sup>1</sup>, There are 101 different classes of food, with 1000 labeled images per class available for supervised training. This project aims at combining object detection algorithms along with Google's pre-trained Inception v3 algorithm and fine tuning the results using data augmentation.

## 2 System Configuration

### 2.1 Hardware Requirements For Local Machine

The implementation is carried out locally using a Jupyter Notebook and the Python version used is 3.9.

**Operating System:** Mac Os

**Processor:** Intel Core i7-7300H @ 3.1 GHz

**RAM:** 16 GB

**Hard drive:** 512 GB SSD

**GPU:** AMD 1650 Ti

### 2.2 Software Requirements

The project implementation is carried out in the latest version of Python 3.8 and the Jupyter notebook interface is used for visually representing the output of the deep learning models. Inception v3 models are tuned based on various hyperparameters and the results are concluded in the Jupyter master notebook. An user interface is built using ReactJs, Javascript, HTML and CSS for aiding the end users to upload images and test the results of the deep learning model. The packages included in the project are listed as follows:

1. Python 3.8
2. React JS
3. HTML

---

<sup>1</sup>Food Dataset:<https://www.kaggle.com/kmader/food41>

4. CSS
5. JavaScript
6. Jupyter Notebook
7. Visual Studio Code Editor
8. TensorFlow
9. Keras
10. pandas
11. matplotlib
12. os
13. urllib
14. json
15. tqdm
16. opencv

### 3 Dataset Description

The dataset for this project is generated from Kaggle <sup>2</sup> and it is an open source public dataset available for research purposes. The dataset has 101 categories of food items and each category is labelled with a minimum of 1000 images. The dataset has been further expanded in size by subjecting the dataset to a series of data augmentation steps.

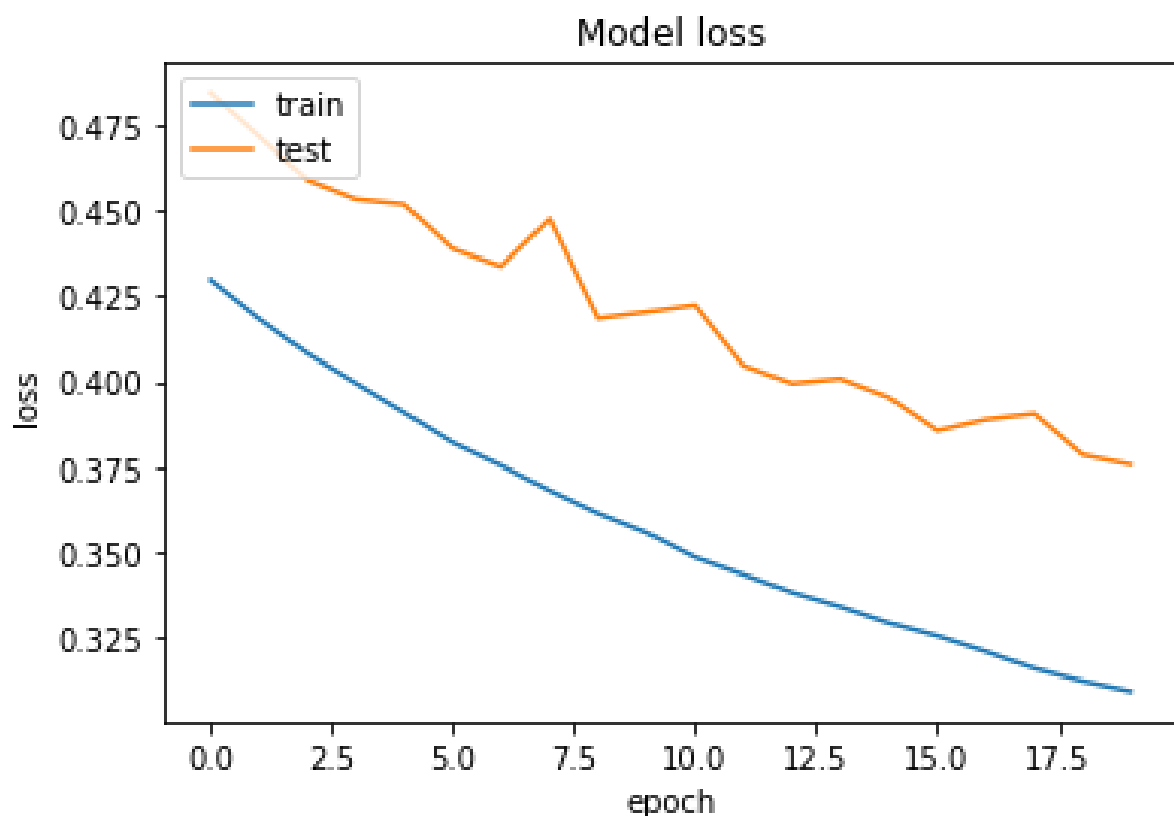


Figure 1: Model Loss Comparison Metrics

---

<sup>2</sup>Food Dataset:<https://www.kaggle.com/kmader/food41>

## 4 Environment Setup

The environment setup has been carried out locally and the local environment needs to be setup by installing a series of softwares and libraries. Beginning with the installation of python 3.8 followed by Anacondas and Jupyter notebook for testing, training and tuning the deep learning models and analyzing them. The Jupyter notebook makes use of the Tensorflow library and the GPU needs to be computationally intensive to handle deep learning and neural networking of the images. For the implementation of user interface, install NodeJS and npm which are library management providers and Visual Studio Code is used as the editor for implementing the frontend UI medium in ReactJS.

For running the Jupyter notebook, navigate to the master notebook and run the entire master notebook to view the results. To run the frontend setup, go to the command prompt or terminal and type `npm install` to install all the dependencies libraries. Then run `npm start` to get the application up and running on the browser. The end users can now view the application running on the local port 8080 and test the deep learning model by uploading test images.


## 5 Training and Implementation of Model with TensorFlow

The data is transformed to a standard resolution and trimmed to filter out unwanted noise and disturbance. The 3 dimensional images are transformed into 2 dimensional images for faster processing. The models are built by using the combination of different hyperparameters and learning rate schedulers to fetch the most successful models based on InceptionNet v2 architecture. The Figure 4 shows the model running and identification of a person in the Azure Machine Learning workspace. This is a workspace we can build the model by creating Jupyter Notebooks.





The 5,6,7 shows the pipeline creation for this MLOps process. The Azure Machine Learning has the inbuilt feature of MLOps (Machine Learning Operation) that can be accessed by the options Azure ML Pipeline. The Pipeline can be accessed by the trained model with the library *"azureml"* package in our code. The running pipeline is shown clearly in the second and third figures that are clicked to view the experiment outputs and also if the code is changed in the workspace and if you run all the model building pipeline will trigger to create the new endpoint that can be accessed by the web application to implement the model in web application.


The Completely trained model will generate a REST API, Through this REST API, you can easily access the model to test and validate. This model-trained end-point API can be accessed with the flask in the back end to easily Validate the authentication process of a Masked Face person. The Figure 8 shows the end-point that is generated for our model and is integrated with the web application.

Finally, we will discuss the model integration part with the web application. Previous figures show the implementation and model building with Azure Cloud Service. The model was built and integrated with the MLOps part with the help of the Azure Machine Learning service. Now we see how to run the developed web application model with flask. From the figure 2, we see the environment setup we made in our local machine to run the




EXPLORER

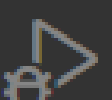




> .vscode

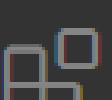


> images



> model

≡ model\_dl.h5



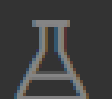
> static

> css

# main.css

> js

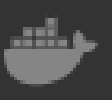
JS main.js




> templates

<> base.html


<> index.html




> Test




> Train




amlrun.py




app.py



MaskedFR\_Model.ipynb



requirements.txt



train.py

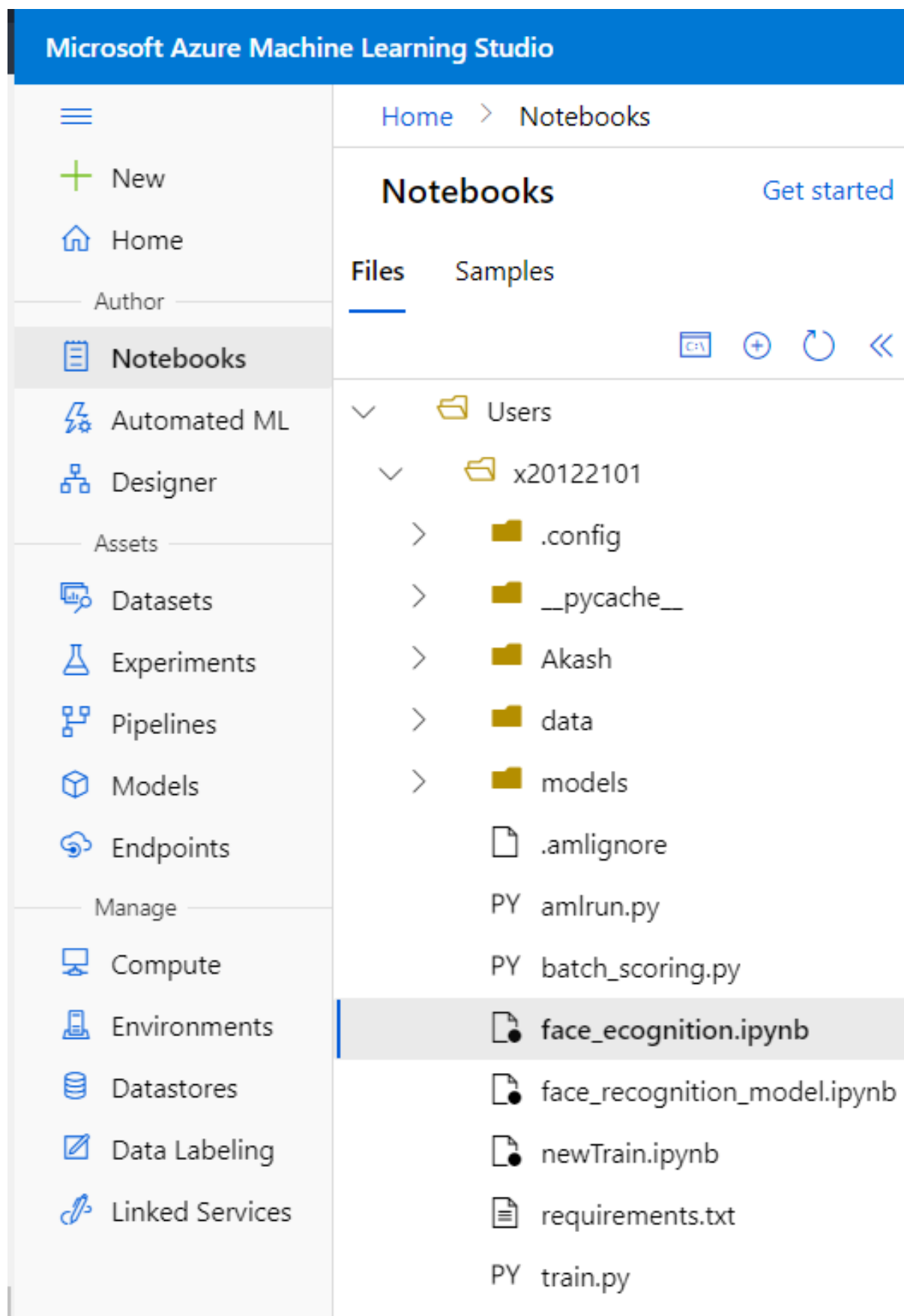


Figure 3: Folder Structure Jupyter Notebook in Azure Machine Learning Workspace



6



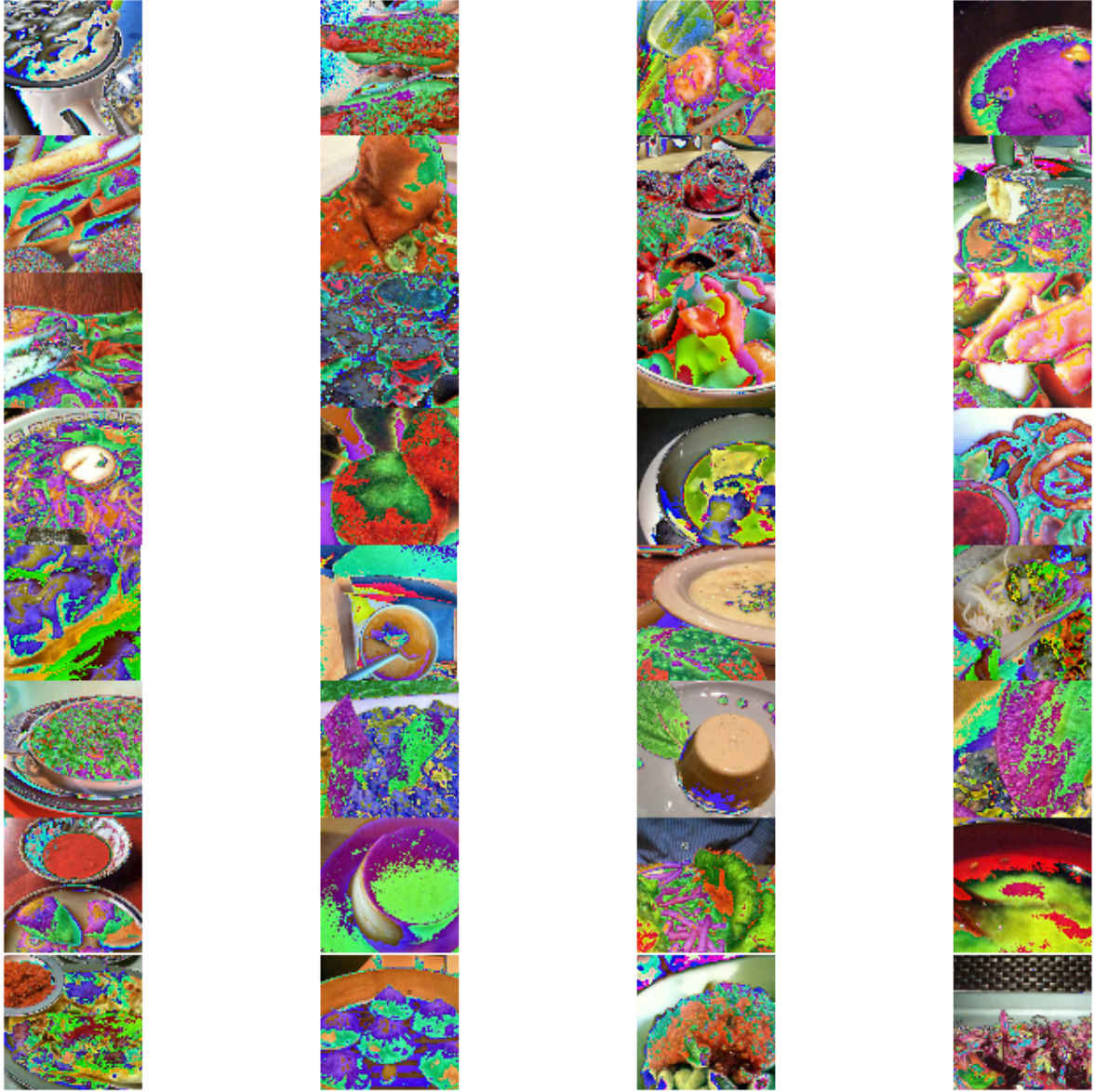


Figure 5: Object Detection of Food Categories

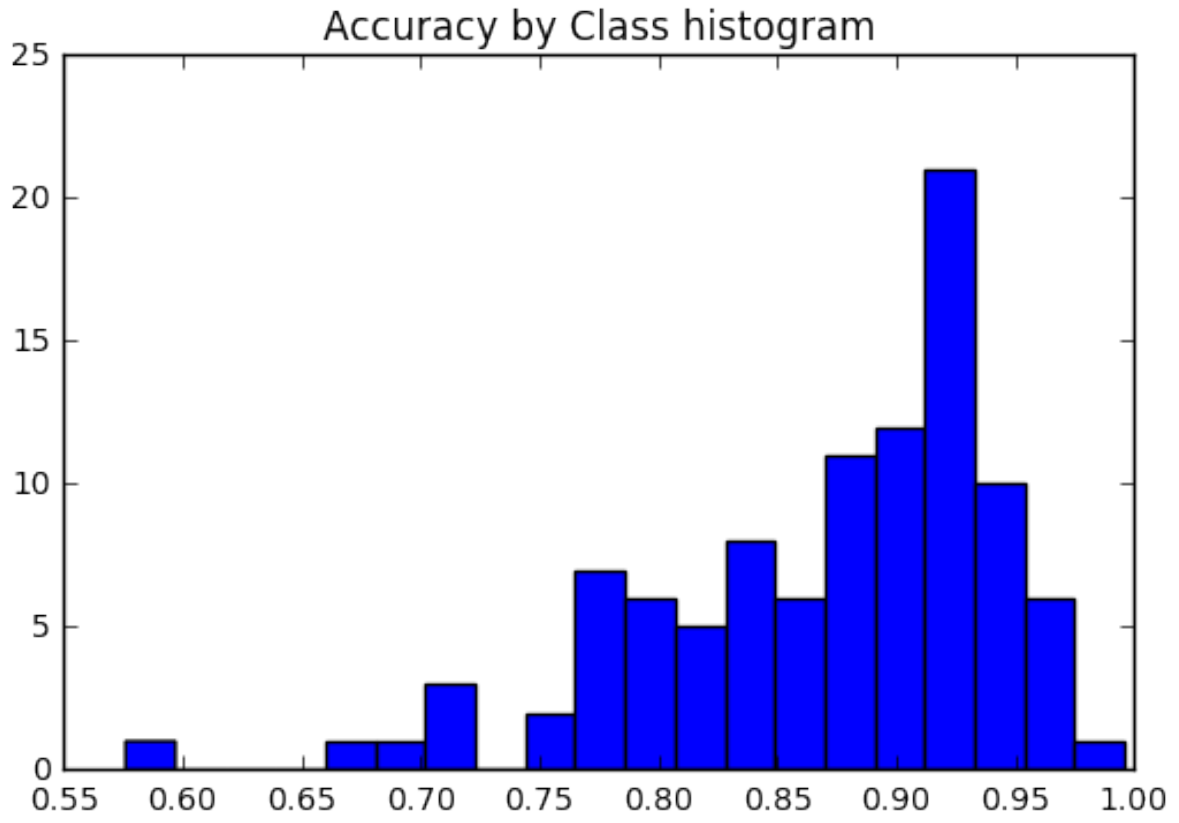


Figure 6: Accuracies Based on Categories

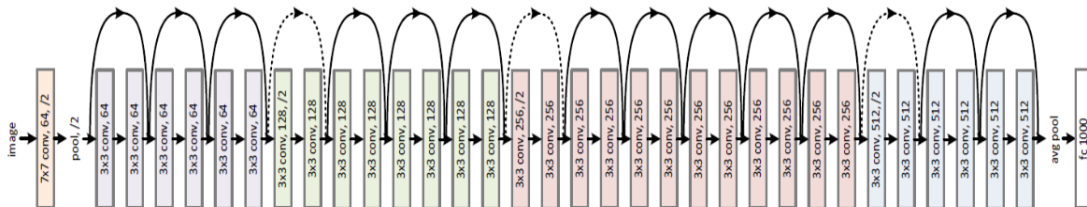


Figure 7: Inception Net v3 architecture

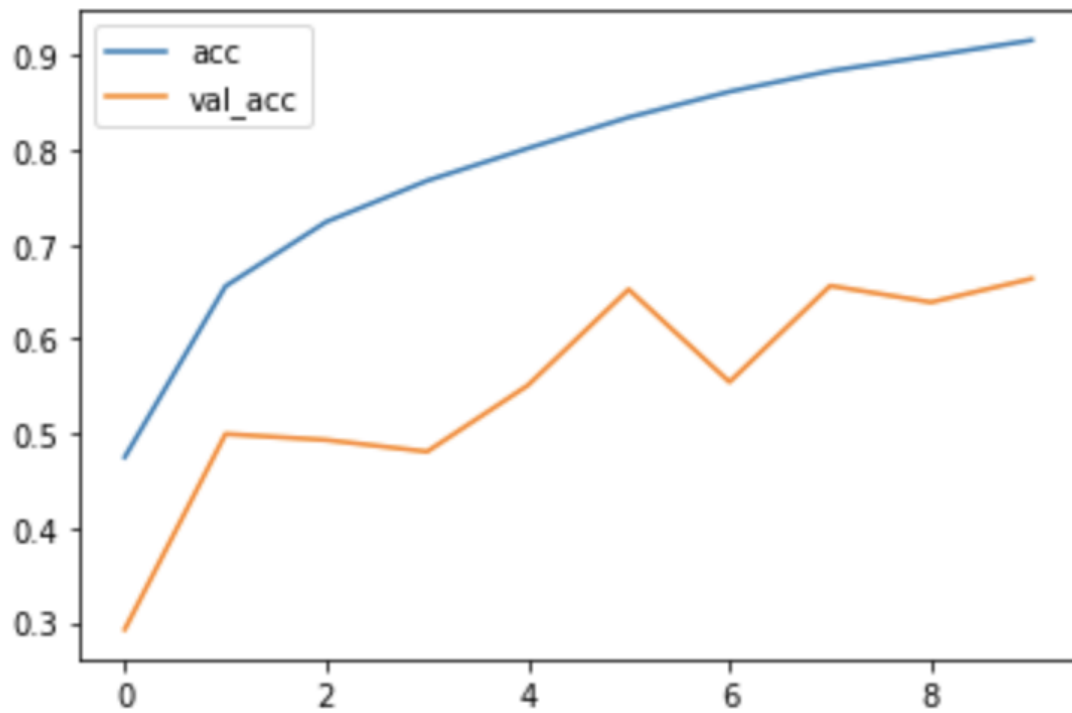


Figure 8: Comparison of Validation and Loss Metrics across Models

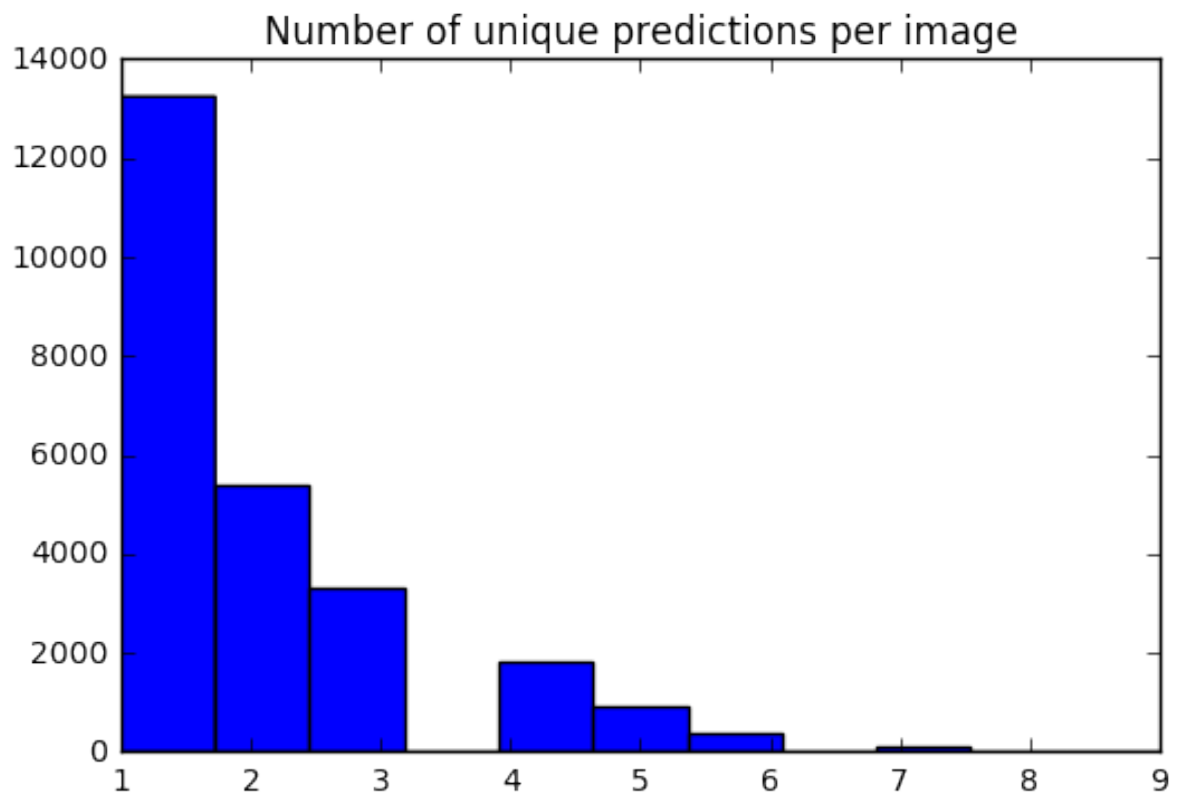


Figure 9: Unique Prediction Categories