

# Food Dish Recognition

CSC4101 Artificial Intelligence

By:

Muhammad Ahsan (1912310)

Shayan Ali Mankani(1912322)

Food logging is considered one of the most data-collecting activities that users do with their smartphones (taking snapshots of their meals, sharing on social media, etc.). There could be ease of use such as taking a picture and getting to know what type of dish it is.

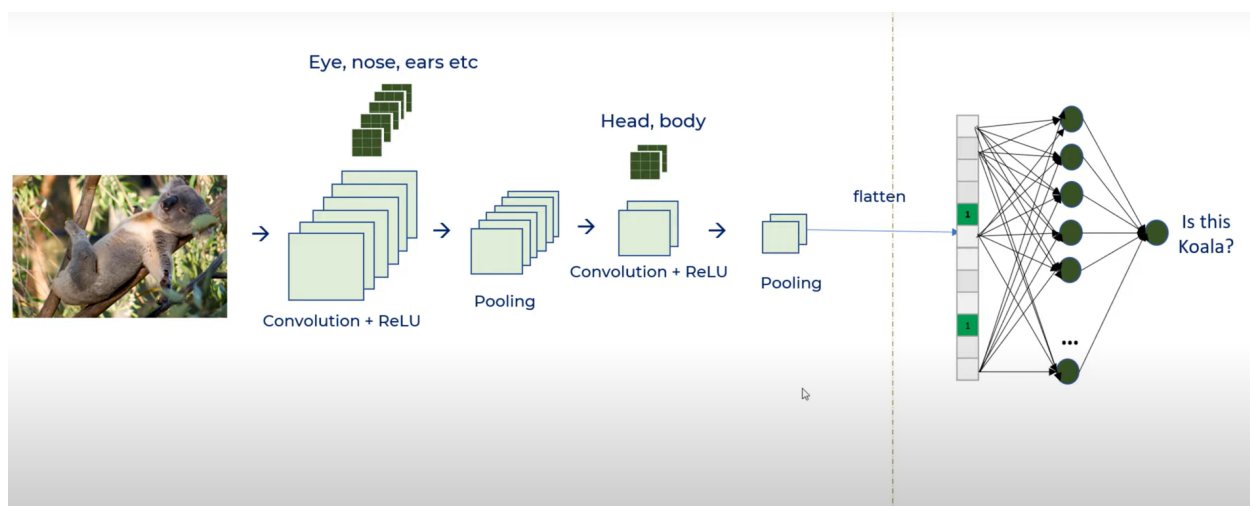
The idea behind the project is to make people, especially foreigners familiar with the Pakistani dishes. It will be able to recognize the dishes by detecting the images.

It is a UI based project with the feature of capturing images and through the means of algorithm it will detect the Pakistani dishes and also indicate the name of the dish. The UI of the project is done through flask framework, with the webcam integration of 720p.

Flask is a web framework, it's a Python module that lets you develop web applications easily. It has a small and easy-to-extend core: it's a microframework that doesn't include an ORM (Object Relational Manager) or such features. It does have many cool features like url routing, template engine. It is a WSGI web app framework.

The Flask framework was more than enough to assist with the features that we required for this project.

The field of machine learning has taken a dramatic twist in recent times, with the rise of the Artificial Neural Network (ANN). These biologically inspired computational models are able to far exceed the performance of previous forms of artificial intelligence in common machine learning tasks. One of the most impressive forms of ANN architecture is that of the Convolutional Neural Network (CNN). CNNs are primarily used to solve difficult image-driven pattern recognition tasks and with their precise yet simple architecture, offer a simplified method of getting started with ANNs.



The project heavily relies on a Convolutional Neural Network (CNN) for the tasks of detecting and recognizing food images. The model that has been opted is Efficient Net. CNN is a great approach to deep learning and with extensive research, deep learning seems to be a powerful technique for image recognition.

EfficientNet is a convolutional neural network architecture and scaling method that uniformly scales all dimensions of depth/width/resolution using a compound coefficient. Unlike conventional practice that arbitrarily scales these factors, the EfficientNet scaling method uniformly scales network width, depth, and resolution with a set of fixed scaling coefficients.

If we talk about the implementation in python, we are using keras and tensorflow. Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research.

Keras reduces developer cognitive load to free you to focus on the parts of the problem that really matter. Simple workflows should be quick and easy, while arbitrarily advanced workflows should be possible via a clear path that builds upon what you've already learned. TensorFlow 2 is an end-to-end, open-source machine learning platform. You can think of it as an infrastructure layer for differentiable programming. It combines four key abilities:

- Efficiently executing low-level tensor operations on CPU, GPU, or TPU.
- Computing the gradient of arbitrary differentiable expressions.
- Scaling computation to many devices, such as clusters of hundreds of GPUs.
- Exporting programs ("graphs") to external runtimes such as servers, browsers, mobile and embedded devices.

Keras is the high-level API of TensorFlow 2: an approachable, highly-productive interface for solving machine learning problems, with a focus on modern deep learning. It provides essential abstractions and building blocks for developing and shipping machine learning solutions with high iteration velocity.

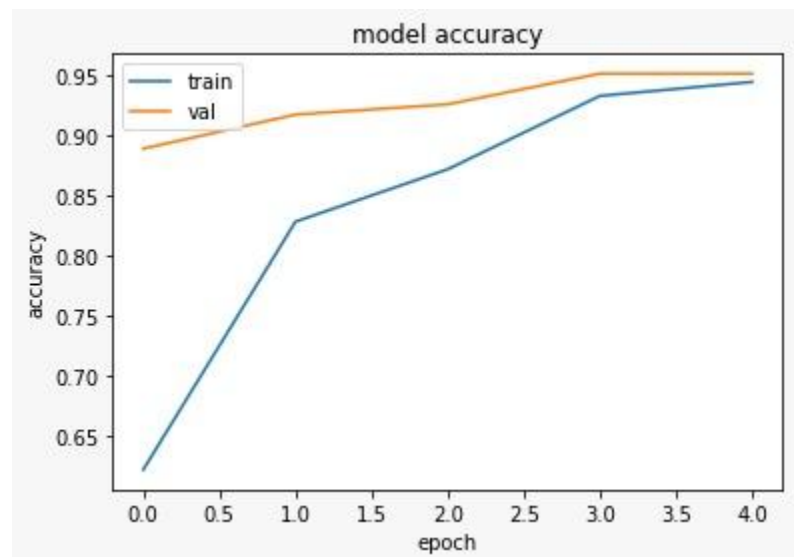
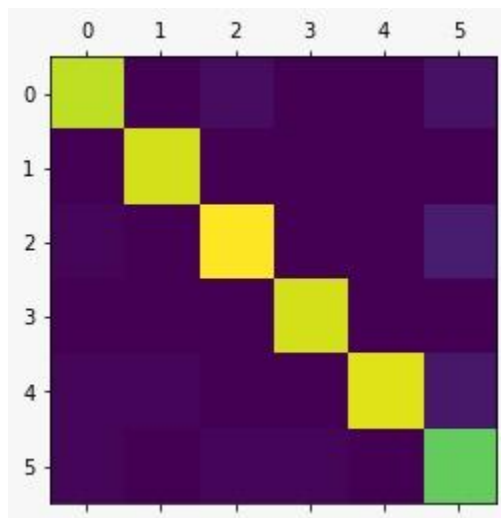
Keras empowers engineers and researchers to take full advantage of the scalability and cross-platform capabilities of TensorFlow 2: you can run Keras on TPU or on large clusters of GPUs, and you can export your Keras models to run in the browser or on a mobile device.

So in conclusion, we are training our keras model on Google Collab and then we are exporting that model on our local network and making a flask web application connected with that model and using a webcam api to connect our webcam to the application. So as this we are getting a trained model with a flask web application connected to it.

Apart from the algorithm, the major element is the clean dataset.

The hardest challenge was to achieve the target of collecting the quality data in a short period of time. We had to use all necessary methods to collect data as soon as possible such as taking pictures of our own, scraping through large food journals and websites to fulfill our dataset requirements. Our goal was to collect 15 dishes each having around 300+ images but due to visualization limitations we had to narrow down to selecting 6 dishes each having 180 images and this was after we did all the filtration and correction of the raw dataset we had collected during the project. The representation of our data is like we have divided the dataset in 3 categories (Training, Evaluation and Validation). The dishes are stored into each of the category folders.

With Pakistani dishes, which most of them seem kind of similar and not having enough data in hand, to attain the accuracy was a challenge. The reasons we had challenges with the accuracy is related to the texture and colors of our Pakistani dishes. The garveys and rice dishes have similarity between them and how CNN works is related to the color and geometry so predicting the correct dish would be challenging for our model since majority of the dishes have same colors. For example we tested between karahi and nihari, every time it either predicted karahi or nihari that's because both have similarities in terms of texture and color at some point. This is prioritized by the CNN as mentioned above, however before finalizing to 6 dishes we did test on 10 dishes but as explained above we faced similar problems related to predictions, there could be many reasons for this but however we had to narrow down to less dishes to get our model to predict precise and accurate. We trained our model on 20, 10 epochs on Google's Collab platform which GPU's power helped us train our model much faster and better.



We did manage to get 96% accuracy and 16% loss. To further test it we had been conducting tests in real time to see whether it predicts or not, however it does predict as we have expected.

In conclusion, our project is a Pakistani food dish deductor which takes an image of a dish, predicts what it is. The project uses a pre-trained CNN model which we trained on Google Collab's platform using a CNN algorithm. Then this model is used on a flask web app for better interface usage.

