

1 Final Training Results:

In the second deliverable we hadn't implemented our own model yet and had decided to split our dataset 8:2 for training and validation.

However, we realized we also needed a split for testing and hence decided to split the data 8:1:1 for training, validation and testing.

As for the model, we also changed the model from the one we used in deliverable 2. For deliverable 2 we weren't sure about how to implement our own CNN and hence decided to use an automatically trained model in ML.Net. However for our final model we used tensorflow.keras to implement our own model.

To make sure that the model is insensitive to image transformations, we used an image data generator to randomly transform images for our training and validation sets.

Our final results have decently improved from the last deliverable when we had an accuracy of 59 percent. However with our current model, we have been able to achieve a validation accuracy of 80.75% with a validation loss of 0.6260.

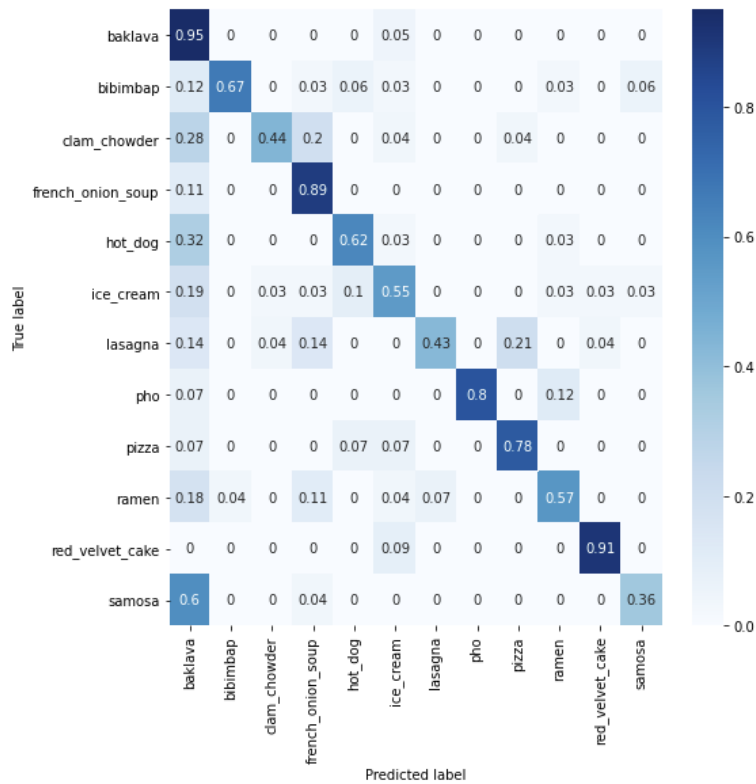


Figure 1: Confusion matrix for validation with 2048 images

2 Final demonstration proposal :

For the final project we hope to integrate our model into an iOS app. To do this, we will utilize Objective-C. We have begun work on the frontend for this app however we still have to work on the backend for the app. We will be working on this in the following weeks to wrap up our project. Through trial and error we will make a great app!



Figure 2: The Prototype