Hongjun, Saumyaa, Kevin

Deliverable 2

Problem Statement:

- We are implementing an image classifier for food items.
- Our implementation will use CNNs (Convolutional Neural Networks) to identify the item of food and then we will use the database of recipes to give back a general ingredient list of the identified food item.

Data Processing:

- For our dataset we are hoping to use a dataset with 101 food categories and 1000 images in each category.
- The dataset is already well labelled however there is a variation in the sizes of the pictures.
- To process these we re-scaled the images to 255 for the short side and cropped them to 255x255.
- We then normalized the images according to values that we obtained from the internet[1].

```
# Credit: https://towardsdatascience.com/a-beginners-tutorial-on-building-an-ai-image-classifier-using-pytorch-6f85cb69cba7

transformations = transforms.Compose([
    transforms.Resize(255),
    transforms.CenterCrop(255),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
])

train_set = datasets.ImageFolder("/content/drive/MyDrive/MAIS202Data/food-101-data/images", transform=transformations)
train_loader = torch.utils.data.DataLoader(train_set, batch_size=32, shuffle=True)
```

Machine Learning Model:

- We have decided to use convolutional neural networks for our project. Since we are not familiar with neural networks, we followed a tutorial[1] to preprocess our data to use in a CNN.
- Once we are familiar with neural networks, we will be able to make more informed decisions about the hyperparameters.
- For testing we used an auto ML library, which automated the testing for us.
- As for our data split, we decided to use 80 percent for training and 20 percent for validation as a starting point.

Preliminary Results:

- Since we are not familiar with neural networks, we used ML.NET to do some auto ML, where we got an accuracy of 59% using a predesigned, automatically trained CNN.
- Clearly the predesigned model does not have a very high accuracy and so is not a feasible choice.

- We also think that the low accuracy may be linked to the low quality of our training and testing dataset. In the future we have decided to clean up our dataset and get rid of these low quality images.

Next Steps:

- Currently we are using a predesigned CNN but in the future we would like to implement this ourselves using PyTorch.
- We plan to work on the frontend and backend of the project before we learn more about neural networks.

References:

[1] Wu, A. (2019, July 31). A beginner's tutorial on building an Ai image classifier using PyTorch. Retrieved February 22, 2021, from https://towardsdatascience.com/a-beginners-tutorial-on-building-an-ai-image-classifier-using-pytorch-6f85cb69cba7