**Problem Statements**:

Imagine that you are a Recipe Author at Cookpad. For an upcoming event you feel motivated to create a dish that is novel and is inspired by your love of world food. After much deliberation you are struck by the simple idea of combining your two favorite light snacks, Sushi and Sandwiches. You are convinced that this dish is unique and decide to call it “Sushidi”. However, to conduct some due diligence you decide to check the Cookpad platform for examples of Sushidi, just to be sure. Based on the provided [dataset](http://research.us-east-1.s3.amazonaws.com/public/sushi_or_sandwich_photos.zip) please develop an automated method or workflow to **analyse the contents and output a list of filenames that could potentially be considered as “Sushidi”**. Your approach can use any area of machine or deep learning in order to accomplish that task.

To classify sushi and sandwich images I have used various image classification models of Tensorflow and Keras. I have tried multiple networks with various models which are described below.

1. **VGG16** model with multiple combination of networks.
2. **ResNet50** model with multiple combination of networks.
3. **VGG19** model with multiple combination of networks.
4. **InceptionV3** model with one combination of networks
5. **MobileNetV2** model with one combination of networks

To train the models, I split the images into train (85%) and test (25%) datasets.  I also implement a data augmentation function where the input images are randomly flipped horizontally, sheared, zoomed and rotated.

1. **VGG16 model implementation**: I have implemented this model with 3 different networks. After comparing their accuracies, I have selected best network and commented the code for the remaining two networks. I have kept the code in comment just for the reference. I got the final accuracy (75.49 - 83.99 %) for this model.
2. **ResNet50** **model implementation:** I have implemented this model also with 3 different networks. After comparing their accuracies, I have selected best network and commented the code for the rest two networks. I have kept the code in comment just for the reference.

I have got the final accuracy (52.49 – 66.00 %) for this model.

1. **VGG19 model implementation:** I have implemented this model with 2 different networks. After comparing their accuracies, I have selected best network and commented the rest two networks code. I got the final accuracy (60.00 – 77.99 %) for this model.
2. **InceptionV3 model implementation:** I have implemented only one network here as I got to know that which networks is giving me better accuracy. I got the final accuracy (67.50 – 79.00 %) for this model.
3. **MobileNetV2 model implementation:** I have implemented only one network here as I got to know that the best networks which is giving me good accuracy. I got the final accuracy (82.99 – 88.99 %) for this model.

Following parameters are constant while performing the various model implementation.

nb\_epoch = 20, (img\_width, img\_height) = 150, 150,

input\_shape = (150, 150, 3) and batch\_size = 32

The final network which I have selected to classify these images is as below:

For the first layer, I am using flatten and dense layer. For the hidden layer I am using Relu as activation function, then dropout and dense layer.  A dropout probability of 0.5 is applied during the training phase between the fully connected layers. I have applied sigmoid activation function at the output layer. Sigmoid function will quash the output values to a range between 0 and 1 making it suitable for binary image classification problems like this one.

**Important Observation**: I found that all the models are giving similar kind of confusion matrix as well as classification report. I am getting values only for True positive, False positive whereas I am not getting any value for True Negative and False Negative. I also found that True positive and False positive values are same for all the models and therefore Racall is coming as 1.00 and precision as 0.5 for class 0 and Racall is coming 0.00 and precision as 1.00 for class 1.

**Conclusion about the Model:** After comparing accuracy of the all the models, the best accuracy was given by MobileNetV2. To compare accuracy in better way, I have kept the networks same for all the models. All the hyperparameters for all the models are same.

Hence, I modify MobileNetV2 model further to get the required results (to predict and classify the images are sushidi).

**Development Environment**: I have used google Colab as a development environment where I have also used GPU to train the model.

**Suggested improvements:**

1. Gather more data to train the model
2. Consider other objects also which are present in the images. I could have isolated other objects such as plate, pan, chopstick, etc. from the images while doing sushidi image classification. This could have improved the performance of models.
3. Implementation of 2nd steps could have also improved the confusion matrix and classification report parameters.