**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 various image classification models of Tensorflow and Keras have been used. Multiple networks with various models that are tried for this problem 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 have split images into train (85%) and test (25%) datasets.  Additionally, data augmentation function is implemented where the input images are randomly flipped horizontally, sheared, zoomed and rotated.

1. **VGG16 model implementation**: This model is implemented with 3 different networks. After comparing their accuracies, best network has been selected for this model. The final accuracy for this model lies between **(75.49 - 83.99** **%)**. The remaining code (for other two layers) is commented and kept there just for the reference.
2. **ResNet50** **model implementation:** This model is also implemented with 3 different networks. After comparing their accuracies, best network has been selected for this model. The final accuracy for this model lies between **(52.49 – 66.00 %).** The remaining code (for other two layers) is commented and kept there just for the reference.
3. **VGG19 model implementation:** This model is implemented with 2 different networks. After comparing their accuracies, best network has been selected for this model. The final accuracy for this model lies between **(60.00 – 77.99 %).** The remaining code (for another layer) is commented and kept there just for the reference.
4. **InceptionV3 model implementation:** This model is implemented only with one different network. After doing experiment with above 3 models, I got to know that which networks is giving me better accuracy. Therefore, only the best network is chosen directly for this model. The final accuracy for this model lies between **(67.50 – 79.00 %).**
5. **MobileNetV2 model implementation:** This model is implemented only with one different network. After doing experiment with above 3 models, I got to know that which networks is giving me better accuracy. Therefore, only the best network is chosen directly for this model. The final accuracy for this model lies between **(82.99 – 88.99 %)** for this model.

Parameters that are constant while performing the various model implementation are described below:

*Number of epochs = 20, (Image width, Image height) = 150, 150, Input share = (150,150) and batch size = 32*

**The final network that has been chosen based on their accuracy to classify these images is as below**:

For the first layer, flatten and dense layer are implemented. For the hidden layer Relu as activation function, then dropout and dense layer are used.  A dropout probability of 0.5 is applied during the training phase between the fully connected layers. Sigmoid activation function is applied 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. The model uses rmsprop as an optimiser to minimise binary cross entropy.

**Important Observation:** It has been observed that all the models are giving similar kind of confusion matrix as well as classification report. True positive and False positive values are coming whereas there are no values for True Negative and False Negative. Additionally, found that True positive and False positive values are same for all the models and therefore Racall is coming as 1.00 & precision as 0.5 for class 0 whereas Racall is coming 0.00 & 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** model. To compare accuracy in better way, networks is kept same for all the models. All the hyperparameters for all the models are also kept same.

Therefore, I modified MobileNetV2 model further to get the required results (to predict and classify sushidi images).

**Development Environment**: Google Colab as a development environment have been used where GPU is used 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.