Zero Deforestation Mission

A first attempt Image classification model

Authored by Gerald Fraser

Schneider Electric

Introduction & Methodology

- The aim of the project was to develop a image classification model, which from a given dataset, predicts what type of deforestation appears in the image with the objective of early detection of this type of actions in protected lands

- For this first iteration of the image classification model, it was decided to use a "**Transfer learning**" approach.

- InceptionV3, the pre-trained Deep convolutional Neural Network, was used to train the training dataset images. This model was chosen due it being trained on one of the biggest image datasets (ImageNet) and having a high accuracy with that dataset.

  - "imagenet" weights and the layers of the InceptionV3 model are unchanged in order to re-use pretrained parameters.

  - However, the input layer is changed to (224,224)

```
# re-size all the images to this
IMAGE_SIZE = [224, 224]

train_path = 'train_test_data/train'
```

```
# Here we will be using imagenet weights

inception = InceptionV3(input_shape=IMAGE_SIZE + [3], weights='imagenet', include_top=False)
```

Schneider Electric

Further Methodology

```
for layer in inception.layers:
    layer.trainable = False
```

```
x = Flatten()(inception.output)
```

```
# softmax is used for multi-class classification predictions
num_of_classes=train['label'].nunique()
prediction = Dense(num_of_classes, activation='softmax')(x)
```

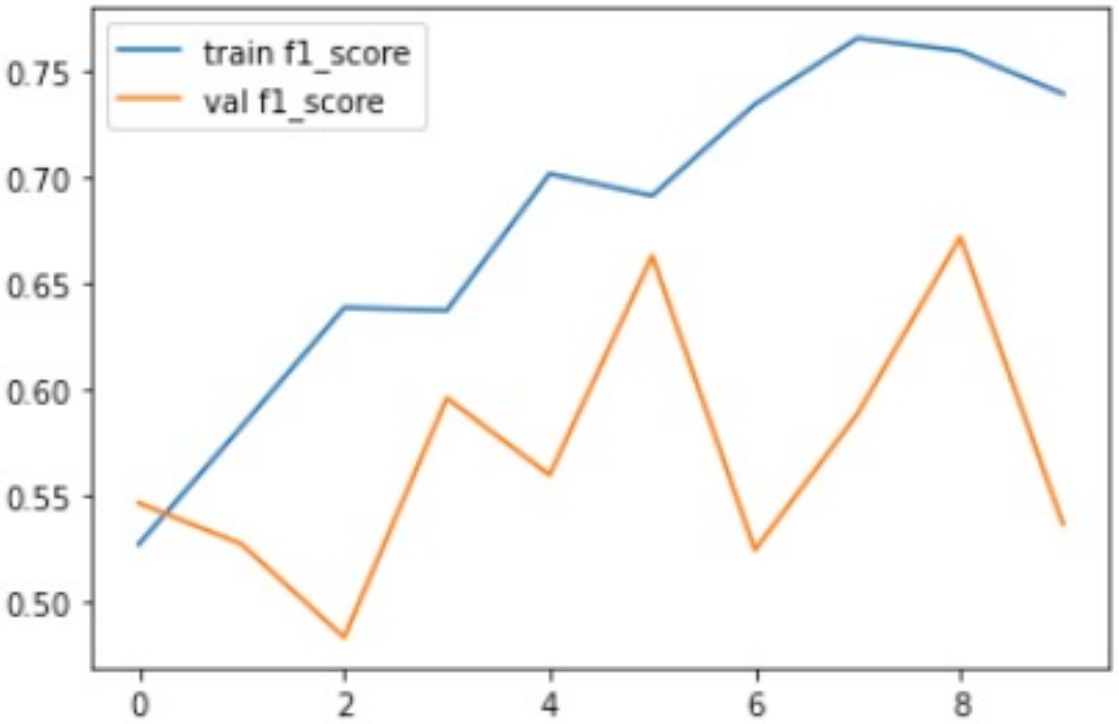
```
# create a model object
model = Model(inputs=inception.input, outputs=prediction)
```

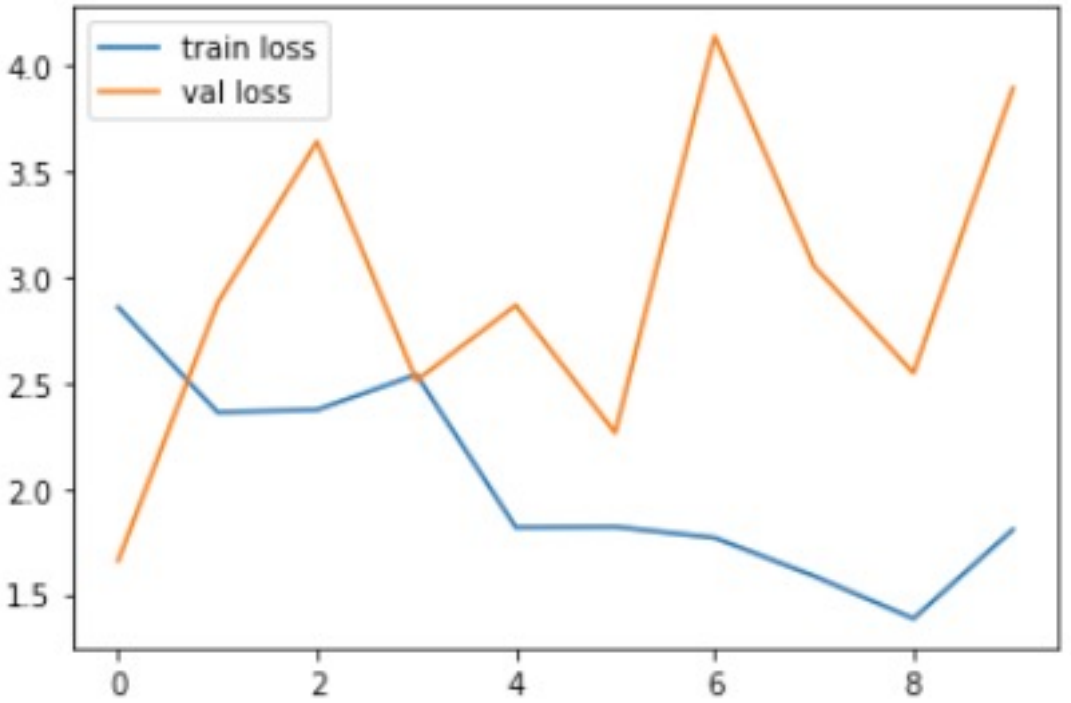
- Other than input and output layer, the rest of the InceptionV3 layers are kept the same
- The output has a flattened layer and then passed onto the dense layer for “softmax” activation which is practical for multiclass predictions.

- The images were pre-processed through data augmentation. Test and training images were rescaled (by 1/255). The Training set only was split 80:20 to generate the validation set for model evaluation.
- Additionally for the training and validation sets, the batch size was set to 16 (for RAM management purposes) and random shuffle and seed were set to true.

Schneider Electric

Results





- As a result of the training and the evaluation, we can see that the model overfitted on the training dataset after just 1 epoch.
- There is increasing loss as the validation f1-score oscillates between 0.5 and 0.65
- The final val f1-score (macro) turned out to be **0.54**

Limitations and Areas to Improve the model

- The features in the “train.csv” and “test.csv” were not used in the InceptionV3 model. Incorporating them may have likely helped the classification.
- Input size of the images for the model were set to (224,224) which are known to be the default input size for VGG16 model. Using the default size of (299,299) might have been preferable for the InceptionV3.
- Better image pre-processing / augmentation may have improved the model evaluation and scoring.
- Adjusting/adding more layers to the CNN model could have led to better performance and improved model evaluation/scoring.
- Using other pre-trained models such as VGG16 & ResNetV2 could have been used to compare and see which models be better for image classification.
- Developing a custom CNN model may also have been an option.