Data Science BCS-7B

Phase 2

Report

Contents:

Members:

17L-4010	Uzair Ahmed Bhatti
17L-4154	Muhammad Saqib Kamran
17L-4238	Muhammad Khuram Niaz
17L-6326	Ali Haseeb

Introduction:

The DS project involved classification of two types of identity cards using a binary classifier followed by detection of credentials mentioned on the cards, through transfer learning.

The project was divided into two phases:

Phase 1 included the annotation and augmentation of images to be used for training the models.

Phase 2 included training the models, then classifying cards and detecting labels on the cards.

Classes/Labels of Cards:

We took two identity cards:

i. Pakistani CNIC card ii. Canadian Passport

For each type of card the number of classes/labels selected were five.

i. Name ii. Identity Number iii. Gender iv. Date of Birth

v. Country

Number of Images:

For each type of card 25 original images were downloaded. After downloading the images, they were annotated using COCO Annotator. We used Docker to run the Coco Annotator on our system. After the images were annotated we used python code for segmentation to augment the images. After augmentation we had a total of 2275 images for each type of card. The .json files for the annotations and augmentation were also generated respectively. These 2275 images of each type of card were used to train the respective models for each card. Moreover, for testing purposes, 5 new images for each type of card were downloaded and augmented which made them a total of 350 images for testing for each type of card.

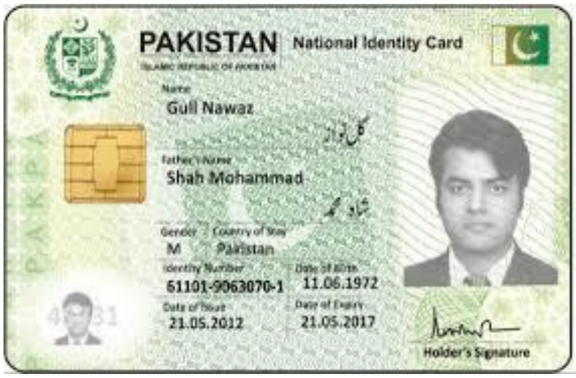
Augmentation Technique:

The augmentation techniques used were flipping the original images as well as rotating each image by 4 degrees. So the total images against 25 images were 2275 for each type of card.

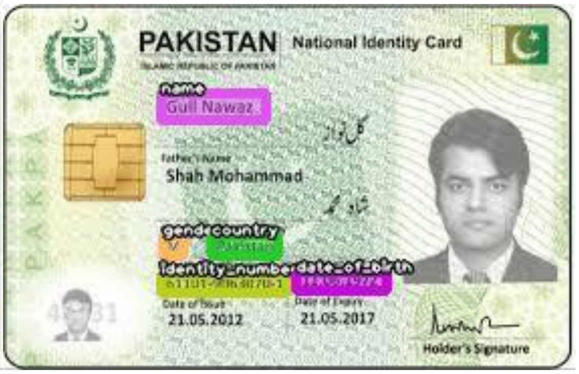
The .json files for both annotation and augmentation are provided within the submission folder.

Sample Images:

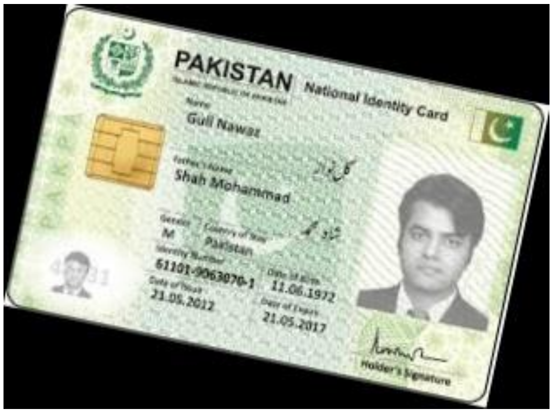
Original Pakistan CNIC



Annotated Pakistan CNIC



Augmented Pakistan CNIC:



PAKISTAN National Identity Card

Name
Gull Nawaz

Father's Name
Shah Mohammad

Gender M
Country of Stay Pakistan

Identity Number
61101-9063070-1

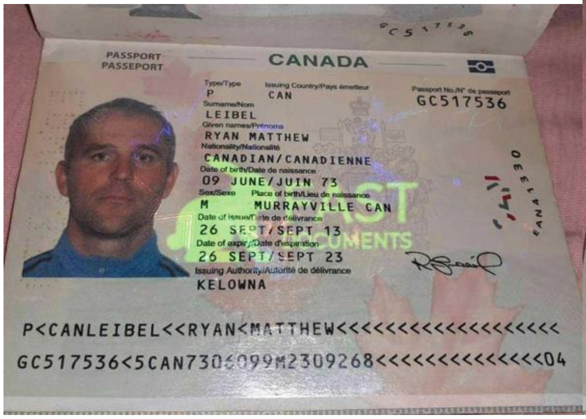
Date of Birth 11.06.1972

Date of Issue 21.05.2012

Date of Expiry 21.05.2017

Holder's Signature

Original Canadian Passport



PASSPORT
PASSEPORT

CANADA

Type/Type P
Issuing Country/Pays émetteur CAN
Passport No./N° de passeport GC517536

Surname/Nom
LEIBEL

Given names/Prénoms
RYAN MATTHEW

Nationality/Nationalité
CANADIAN/CANADIENNE

Date of birth/Date de naissance
09 JUNE/JUIN 73

Sex/Sexe M
Place of birth/Lieu de naissance MURRAYVILLE CAN

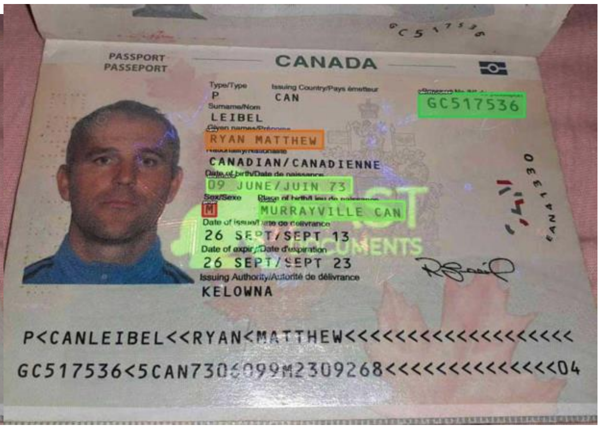
Date of issue/Date de délivrance
26 SEPT/SEPT 13

Date of expiry/Date d'expiration
26 SEPT/SEPT 23

Issuing Authority/Autorité de délivrance
KELOWNA

```
P<CANLEIBEL<<RYAN<MATTHEW<<<<<<<<<<<<<<<<<<<
GC517536<5CAN7306099M2309268<<<<<<<<<<<<<<04
```

Annotated Canadian Passport



PASSPORT
PASSEPORT

CANADA

Type/Type P
Issuing Country/Pays émetteur CAN
Passport No./N° de passeport GC517536

Surname/Nom
LEIBEL

Given names/Prénoms
RYAN MATTHEW

Nationality/Nationalité
CANADIAN/CANADIENNE

Date of birth/Date de naissance
09 JUNE/JUIN 73

Sex/Sexe M
Place of birth/Lieu de naissance MURRAYVILLE CAN

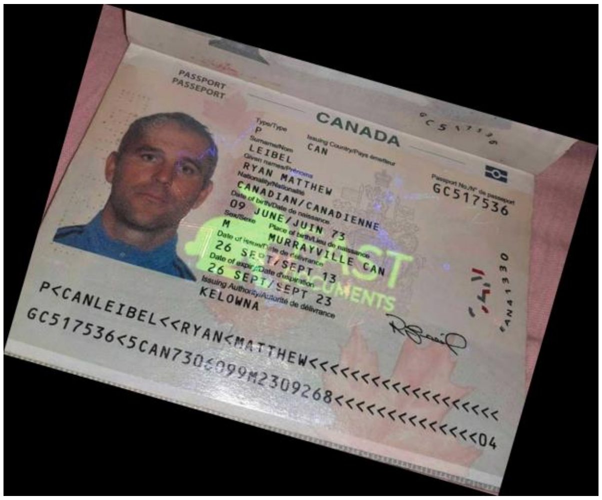
Date of issue/Date de délivrance
26 SEPT/SEPT 13

Date of expiry/Date d'expiration
26 SEPT/SEPT 23

Issuing Authority/Autorité de délivrance
KELOWNA

P<CANLEIBEL<<RYAN<MATTHEW<<<<<<<<<<<<<<<<<<<
GC517536<5CAN7306099M2309268<<<<<<<<<<<<<<04

Augmented Canadian Passport



PASSPORT
PASSEPORT

CANADA

Type/Type
P

Issuing Country/Pays émetteur
CAN

Passport No./N° de passeport
GC517536

Surname/Nom
LEIBEL

Given names/Prénoms
RYAN MATTHEW

Nationality/Nationalité
CANADIAN/CANADIENNE

Date of birth/Date de naissance
09 JUNE/JUIN 73

Sex/Sexe
M

Place of birth/Lieu de naissance
MURRAYVILLE CAN

Date of issue/Date de délivrance
26 SEPT/SEPT 13

Date of expiry/Date d'expiration
26 SEPT/SEPT 23

Issuing Authority/Autorité de délivrance
KELOWNA

```
P<CANLEIBEL<<RYAN<MATTHEW<<<<<<<<<<<<<<<<<<<
GC517536<5CAN7306099M2309268<<<<<<<<<<<<<<04
```

Code:

The notebook containing the code for augmentation has been shared within the submission folder as well as the main code.

Model Training and Class/Label Detection:

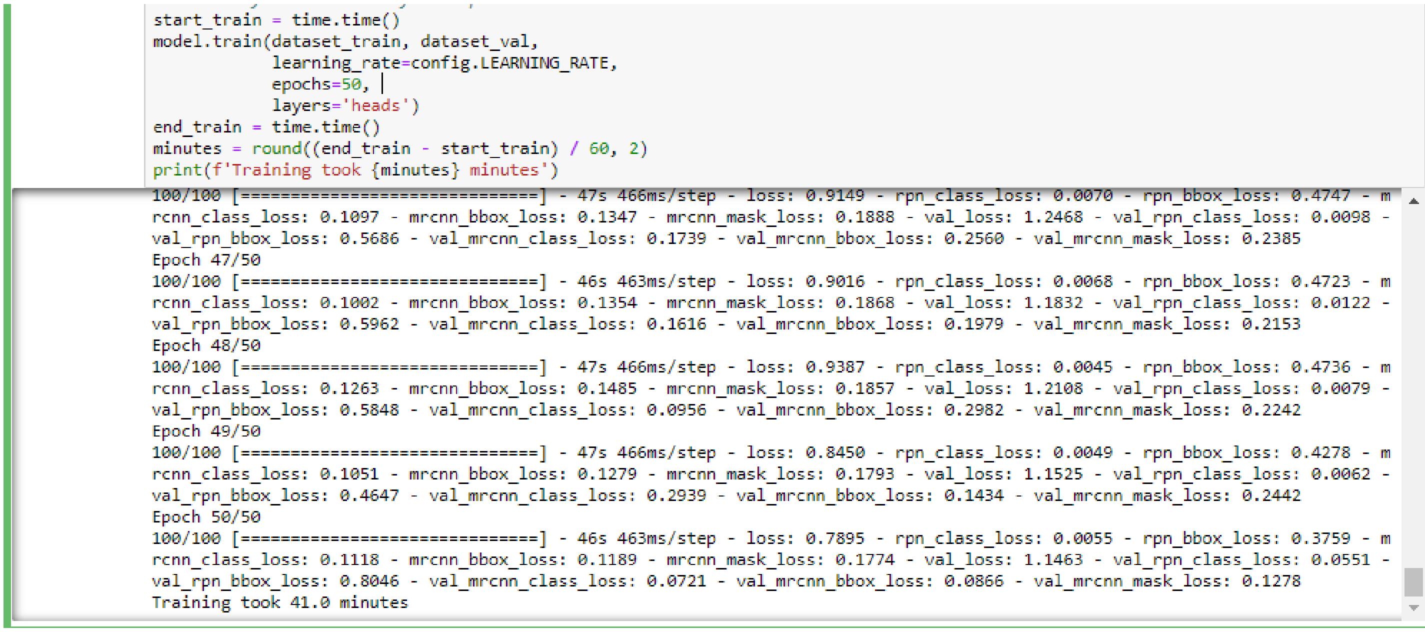
For detection of classes/labels we used Matterport's implementation of Mask R-CNN to train on our dataset of 2275 images of each type of card and then used the trained weights to run inference on new images.

Firstly, using the cloned Matterport's Mask_RCNN we downloaded the pre-trained coco model. Then the configuration for the training of the model was set according to our requirements i.e the min and max resolution of images was set. In addition, the images per GPU were set and the number of classes/labels was set. Moreover, the steps per epoch were also set. In other words, all the configurations were set in the configuration class. We used resnet50. We used Google Colab to train our models using the GPUs provided. The data for the images from the .json files was loaded and then all the annotations were added and then all image's data was added to the dataset. Then using a function the masks for each image were loaded i.e the width, height and instances. Then the paths for the training and validation images were given followed by displaying some sample images from the dataset. Then we moved towards the training of the model. The training model was initialized with COCO and the pretrained weights were loaded.

Training

Training was divided into two stages:

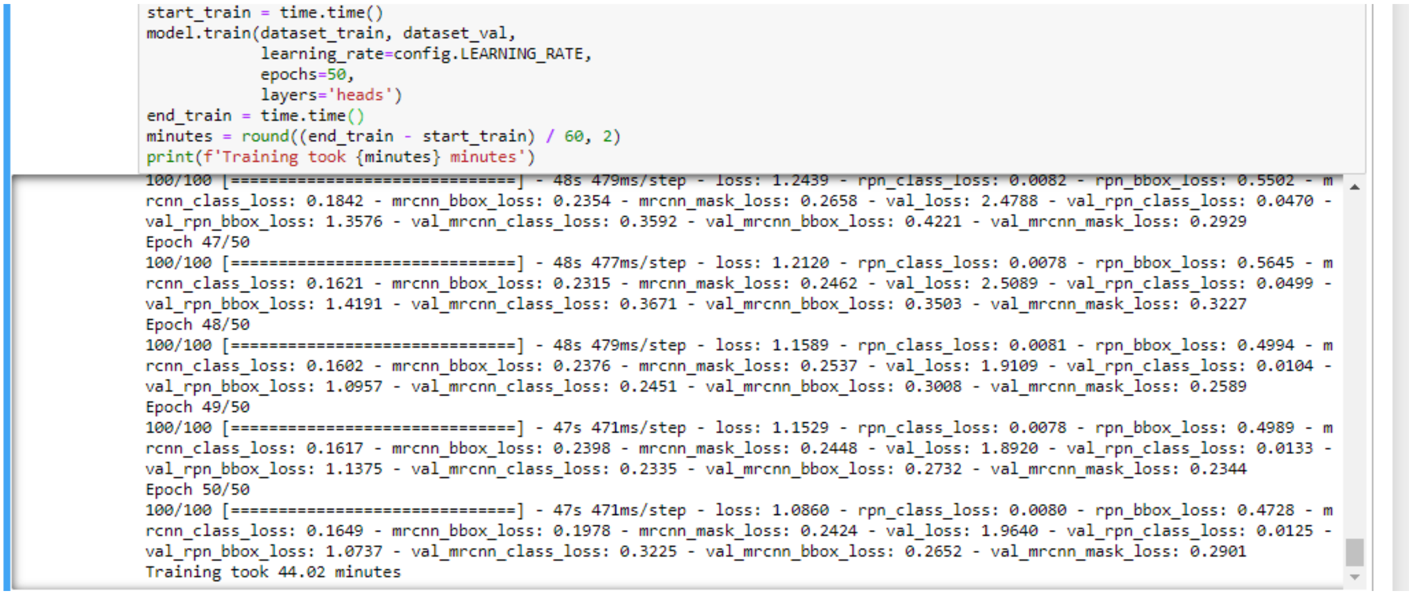
Stage 1: Here we freezed all the backbone layers and trained only the head layer (i.e. the ones that we did not use pre-trained weights from MS COCO). To train only the head layers, we passed layers='heads' to the train() function. The number of epochs used here were 50. The training time was 41 minutes for Pakistan CNIC and 44 minutes for Canadian Passport. The loss was minimized to 0.7 for Pakistan CNIC and 1.08 for Canadian Passport.



```
start_train = time.time()
model.train(dataset_train, dataset_val,
            learning_rate=config.LEARNING_RATE,
            epochs=50, 
            layers='heads')
end_train = time.time()
minutes = round((end_train - start_train) / 60, 2)
print(f'Training took {minutes} minutes')
```

```
100/100 [==============================] - 47s 466ms/step - loss: 0.9149 - rpn_class_loss: 0.0070 - rpn_bbox_loss: 0.4747 - m
rcnn_class_loss: 0.1097 - mrcnn_bbox_loss: 0.1347 - mrcnn_mask_loss: 0.1888 - val_loss: 1.2468 - val_rpn_class_loss: 0.0098 -
val_rpn_bbox_loss: 0.5686 - val_mrcnn_class_loss: 0.1739 - val_mrcnn_bbox_loss: 0.2560 - val_mrcnn_mask_loss: 0.2385
Epoch 47/50
100/100 [==============================] - 46s 463ms/step - loss: 0.9016 - rpn_class_loss: 0.0068 - rpn_bbox_loss: 0.4723 - m
rcnn_class_loss: 0.1002 - mrcnn_bbox_loss: 0.1354 - mrcnn_mask_loss: 0.1868 - val_loss: 1.1832 - val_rpn_class_loss: 0.0122 -
val_rpn_bbox_loss: 0.5962 - val_mrcnn_class_loss: 0.1616 - val_mrcnn_bbox_loss: 0.1979 - val_mrcnn_mask_loss: 0.2153
Epoch 48/50
100/100 [==============================] - 47s 466ms/step - loss: 0.9387 - rpn_class_loss: 0.0045 - rpn_bbox_loss: 0.4736 - m
rcnn_class_loss: 0.1263 - mrcnn_bbox_loss: 0.1485 - mrcnn_mask_loss: 0.1857 - val_loss: 1.2108 - val_rpn_class_loss: 0.0079 -
val_rpn_bbox_loss: 0.5848 - val_mrcnn_class_loss: 0.0956 - val_mrcnn_bbox_loss: 0.2982 - val_mrcnn_mask_loss: 0.2242
Epoch 49/50
100/100 [==============================] - 47s 466ms/step - loss: 0.8450 - rpn_class_loss: 0.0049 - rpn_bbox_loss: 0.4278 - m
rcnn_class_loss: 0.1051 - mrcnn_bbox_loss: 0.1279 - mrcnn_mask_loss: 0.1793 - val_loss: 1.1525 - val_rpn_class_loss: 0.0062 -
val_rpn_bbox_loss: 0.4647 - val_mrcnn_class_loss: 0.2939 - val_mrcnn_bbox_loss: 0.1434 - val_mrcnn_mask_loss: 0.2442
Epoch 50/50
100/100 [==============================] - 46s 463ms/step - loss: 0.7895 - rpn_class_loss: 0.0055 - rpn_bbox_loss: 0.3759 - m
rcnn_class_loss: 0.1118 - mrcnn_bbox_loss: 0.1189 - mrcnn_mask_loss: 0.1774 - val_loss: 1.1463 - val_rpn_class_loss: 0.0551 -
val_rpn_bbox_loss: 0.8046 - val_mrcnn_class_loss: 0.0721 - val_mrcnn_bbox_loss: 0.0866 - val_mrcnn_mask_loss: 0.1278
Training took 41.0 minutes
```

Pakistan CNIC Training

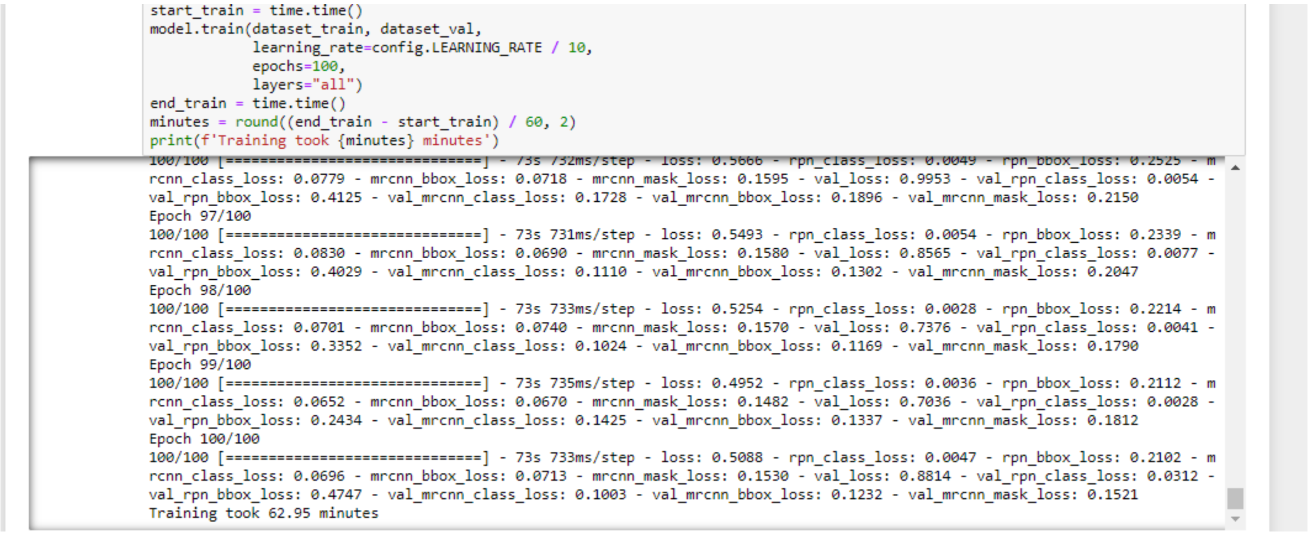


```
start_train = time.time()
model.train(dataset_train, dataset_val,
            learning_rate=config.LEARNING_RATE,
            epochs=50,
            layers='heads')
end_train = time.time()
minutes = round((end_train - start_train) / 60, 2)
print(f'Training took {minutes} minutes')
```

```
100/100 [==============================] - 48s 479ms/step - loss: 1.2439 - rpn_class_loss: 0.0082 - rpn_bbox_loss: 0.5502 - m
rcnn_class_loss: 0.1842 - mrcnn_bbox_loss: 0.2354 - mrcnn_mask_loss: 0.2658 - val_loss: 2.4788 - val_rpn_class_loss: 0.0470 -
val_rpn_bbox_loss: 1.3576 - val_mrcnn_class_loss: 0.3592 - val_mrcnn_bbox_loss: 0.4221 - val_mrcnn_mask_loss: 0.2929
Epoch 47/50
100/100 [==============================] - 48s 477ms/step - loss: 1.2120 - rpn_class_loss: 0.0078 - rpn_bbox_loss: 0.5645 - m
rcnn_class_loss: 0.1621 - mrcnn_bbox_loss: 0.2315 - mrcnn_mask_loss: 0.2462 - val_loss: 2.5089 - val_rpn_class_loss: 0.0499 -
val_rpn_bbox_loss: 1.4191 - val_mrcnn_class_loss: 0.3671 - val_mrcnn_bbox_loss: 0.3503 - val_mrcnn_mask_loss: 0.3227
Epoch 48/50
100/100 [==============================] - 48s 479ms/step - loss: 1.1589 - rpn_class_loss: 0.0081 - rpn_bbox_loss: 0.4994 - m
rcnn_class_loss: 0.1602 - mrcnn_bbox_loss: 0.2376 - mrcnn_mask_loss: 0.2537 - val_loss: 1.9109 - val_rpn_class_loss: 0.0104 -
val_rpn_bbox_loss: 1.0957 - val_mrcnn_class_loss: 0.2451 - val_mrcnn_bbox_loss: 0.3008 - val_mrcnn_mask_loss: 0.2589
Epoch 49/50
100/100 [==============================] - 47s 471ms/step - loss: 1.1529 - rpn_class_loss: 0.0078 - rpn_bbox_loss: 0.4989 - m
rcnn_class_loss: 0.1617 - mrcnn_bbox_loss: 0.2398 - mrcnn_mask_loss: 0.2448 - val_loss: 1.8920 - val_rpn_class_loss: 0.0133 -
val_rpn_bbox_loss: 1.1375 - val_mrcnn_class_loss: 0.2335 - val_mrcnn_bbox_loss: 0.2732 - val_mrcnn_mask_loss: 0.2344
Epoch 50/50
100/100 [==============================] - 47s 471ms/step - loss: 1.0860 - rpn_class_loss: 0.0080 - rpn_bbox_loss: 0.4728 - m
rcnn_class_loss: 0.1649 - mrcnn_bbox_loss: 0.1978 - mrcnn_mask_loss: 0.2424 - val_loss: 1.9640 - val_rpn_class_loss: 0.0125 -
val_rpn_bbox_loss: 1.0737 - val_mrcnn_class_loss: 0.3225 - val_mrcnn_bbox_loss: 0.2652 - val_mrcnn_mask_loss: 0.2901
Training took 44.02 minutes
```

Canada Passport Training

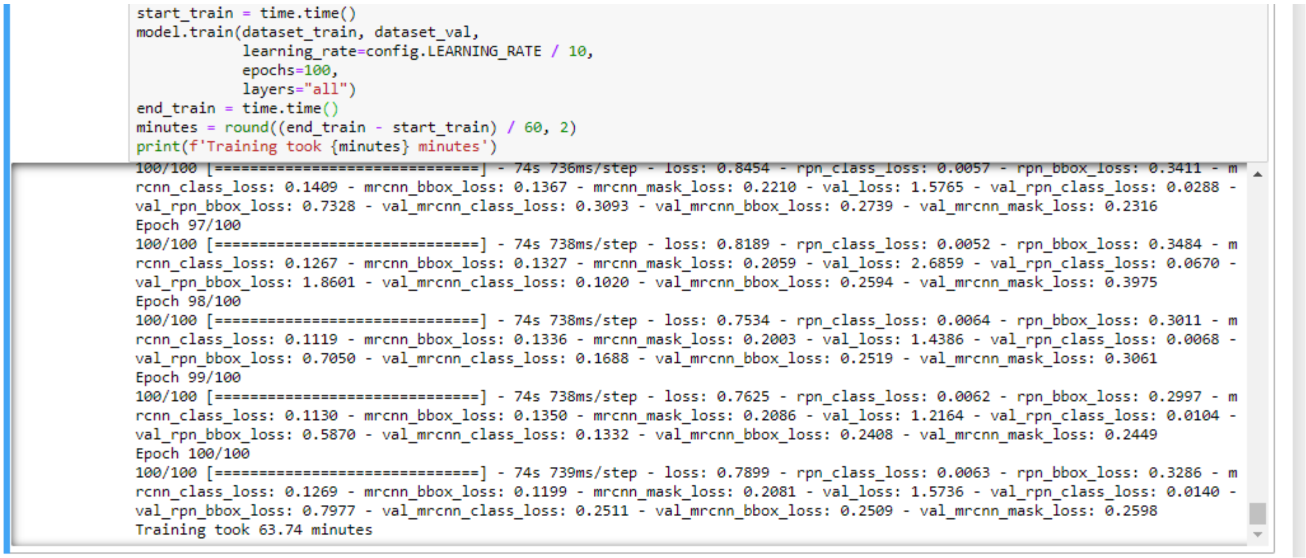
Stage 2: Fine-tune all layers using 100 epochs. The training took about 63 minutes for Pakistani CNIC and 64 minutes for Canada Passport and the loss was minimized to 0.5 for Pakistani CNIC and 0.7 for Canada Passport.



```
start_train = time.time()
model.train(dataset_train, dataset_val,
            learning_rate=config.LEARNING_RATE / 10,
            epochs=100,
            layers="all")
end_train = time.time()
minutes = round((end_train - start_train) / 60, 2)
print(f'Training took {minutes} minutes')
```

```
100/100 [==============================] - 73s 732ms/step - loss: 0.5666 - rpn_class_loss: 0.0049 - rpn_bbox_loss: 0.2525 - m
rcnn_class_loss: 0.0779 - mrcnn_bbox_loss: 0.0718 - mrcnn_mask_loss: 0.1595 - val_loss: 0.9953 - val_rpn_class_loss: 0.0054 -
val_rpn_bbox_loss: 0.4125 - val_mrcnn_class_loss: 0.1728 - val_mrcnn_bbox_loss: 0.1896 - val_mrcnn_mask_loss: 0.2150
Epoch 97/100
100/100 [==============================] - 73s 731ms/step - loss: 0.5493 - rpn_class_loss: 0.0054 - rpn_bbox_loss: 0.2339 - m
rcnn_class_loss: 0.0830 - mrcnn_bbox_loss: 0.0690 - mrcnn_mask_loss: 0.1580 - val_loss: 0.8565 - val_rpn_class_loss: 0.0077 -
val_rpn_bbox_loss: 0.4029 - val_mrcnn_class_loss: 0.1110 - val_mrcnn_bbox_loss: 0.1302 - val_mrcnn_mask_loss: 0.2047
Epoch 98/100
100/100 [==============================] - 73s 733ms/step - loss: 0.5254 - rpn_class_loss: 0.0028 - rpn_bbox_loss: 0.2214 - m
rcnn_class_loss: 0.0701 - mrcnn_bbox_loss: 0.0740 - mrcnn_mask_loss: 0.1570 - val_loss: 0.7376 - val_rpn_class_loss: 0.0041 -
val_rpn_bbox_loss: 0.3352 - val_mrcnn_class_loss: 0.1024 - val_mrcnn_bbox_loss: 0.1169 - val_mrcnn_mask_loss: 0.1790
Epoch 99/100
100/100 [==============================] - 73s 735ms/step - loss: 0.4952 - rpn_class_loss: 0.0036 - rpn_bbox_loss: 0.2112 - m
rcnn_class_loss: 0.0652 - mrcnn_bbox_loss: 0.0670 - mrcnn_mask_loss: 0.1482 - val_loss: 0.7036 - val_rpn_class_loss: 0.0028 -
val_rpn_bbox_loss: 0.2434 - val_mrcnn_class_loss: 0.1425 - val_mrcnn_bbox_loss: 0.1337 - val_mrcnn_mask_loss: 0.1812
Epoch 100/100
100/100 [==============================] - 73s 733ms/step - loss: 0.5088 - rpn_class_loss: 0.0047 - rpn_bbox_loss: 0.2102 - m
rcnn_class_loss: 0.0696 - mrcnn_bbox_loss: 0.0713 - mrcnn_mask_loss: 0.1530 - val_loss: 0.8814 - val_rpn_class_loss: 0.0312 -
val_rpn_bbox_loss: 0.4747 - val_mrcnn_class_loss: 0.1003 - val_mrcnn_bbox_loss: 0.1232 - val_mrcnn_mask_loss: 0.1521
Training took 62.95 minutes
```

Pakistan CNIC fine tuning



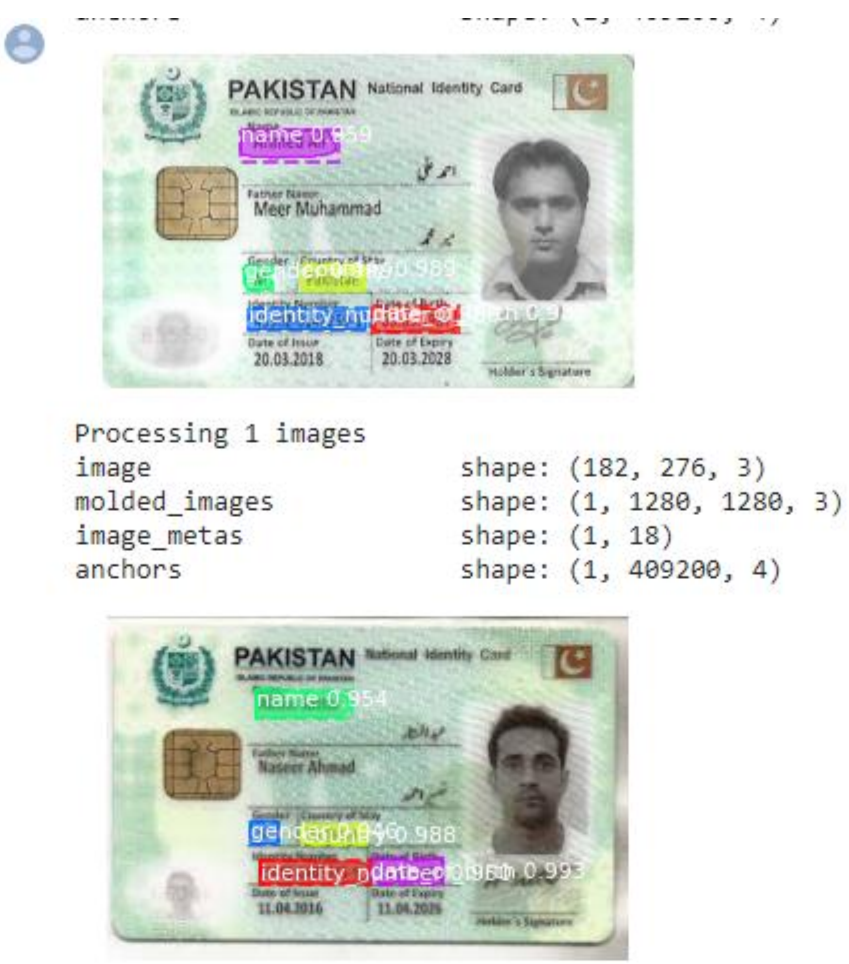
```
start_train = time.time()
model.train(dataset_train, dataset_val,
            learning_rate=config.LEARNING_RATE / 10,
            epochs=100,
            layers="all")
end_train = time.time()
minutes = round((end_train - start_train) / 60, 2)
print(f'Training took {minutes} minutes')
```

```
100/100 [==============================] - 74s 736ms/step - loss: 0.8454 - rpn_class_loss: 0.0057 - rpn_bbox_loss: 0.3411 - m
rcnn_class_loss: 0.1409 - mrcnn_bbox_loss: 0.1367 - mrcnn_mask_loss: 0.2210 - val_loss: 1.5765 - val_rpn_class_loss: 0.0288 -
val_rpn_bbox_loss: 0.7328 - val_mrcnn_class_loss: 0.3093 - val_mrcnn_bbox_loss: 0.2739 - val_mrcnn_mask_loss: 0.2316
Epoch 97/100
100/100 [==============================] - 74s 738ms/step - loss: 0.8189 - rpn_class_loss: 0.0052 - rpn_bbox_loss: 0.3484 - m
rcnn_class_loss: 0.1267 - mrcnn_bbox_loss: 0.1327 - mrcnn_mask_loss: 0.2059 - val_loss: 2.6859 - val_rpn_class_loss: 0.0670 -
val_rpn_bbox_loss: 1.8601 - val_mrcnn_class_loss: 0.1020 - val_mrcnn_bbox_loss: 0.2594 - val_mrcnn_mask_loss: 0.3975
Epoch 98/100
100/100 [==============================] - 74s 738ms/step - loss: 0.7534 - rpn_class_loss: 0.0064 - rpn_bbox_loss: 0.3011 - m
rcnn_class_loss: 0.1119 - mrcnn_bbox_loss: 0.1336 - mrcnn_mask_loss: 0.2003 - val_loss: 1.4386 - val_rpn_class_loss: 0.0068 -
val_rpn_bbox_loss: 0.7050 - val_mrcnn_class_loss: 0.1688 - val_mrcnn_bbox_loss: 0.2519 - val_mrcnn_mask_loss: 0.3061
Epoch 99/100
100/100 [==============================] - 74s 738ms/step - loss: 0.7625 - rpn_class_loss: 0.0062 - rpn_bbox_loss: 0.2997 - m
rcnn_class_loss: 0.1130 - mrcnn_bbox_loss: 0.1350 - mrcnn_mask_loss: 0.2086 - val_loss: 1.2164 - val_rpn_class_loss: 0.0104 -
val_rpn_bbox_loss: 0.5870 - val_mrcnn_class_loss: 0.1332 - val_mrcnn_bbox_loss: 0.2408 - val_mrcnn_mask_loss: 0.2449
Epoch 100/100
100/100 [==============================] - 74s 739ms/step - loss: 0.7899 - rpn_class_loss: 0.0063 - rpn_bbox_loss: 0.3286 - m
rcnn_class_loss: 0.1269 - mrcnn_bbox_loss: 0.1199 - mrcnn_mask_loss: 0.2081 - val_loss: 1.5736 - val_rpn_class_loss: 0.0140 -
val_rpn_bbox_loss: 0.7977 - val_mrcnn_class_loss: 0.2511 - val_mrcnn_bbox_loss: 0.2509 - val_mrcnn_mask_loss: 0.2598
Training took 63.74 minutes
```

Canada Passport fine tuning

Finally, after training, we used inference to test the new images so a class of inference configuration was made and an inference model was made so we could test the images. We are providing some sample detected label pictures. However, completely detected 350 images are shown in the output of the notebook submitted within the submission folder.

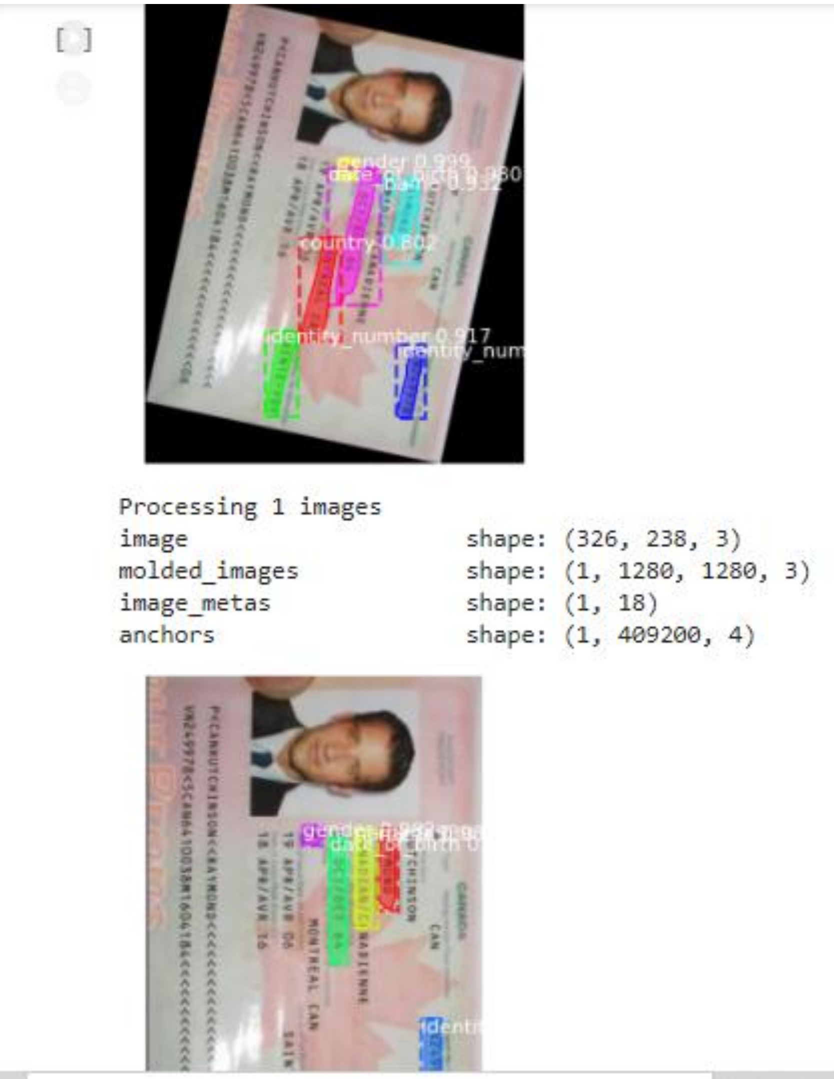
Pakistan CNIC Detection:



```
anchors                  shape: (1, 409200, 4)
```

```
Processing 1 images
image                    shape: (182, 276, 3)
molded_images            shape: (1, 1280, 1280, 3)
image_metas              shape: (1, 18)
anchors                  shape: (1, 409200, 4)
```

Canada Passport Detection:



```
Processing 1 images
image                      shape: (326, 238, 3)
molded_images              shape: (1, 1280, 1280, 3)
image_metas                shape: (1, 18)
anchors                    shape: (1, 409200, 4)
```

Accuracy:

Testing was done on 350 images for each type of card and the labels for Pakistan CNIC were detected with an accuracy greater than 90% with a confidence of 0.90. However, the labels of Canada Passport were detected with an accuracy close to 90% with a confidence of 0.80. For Pakistani CNIC almost every class was detected accurately.

Classification (Binary classifier for classification of cards):

In our project, we firstly classified the images and then detected the labels.

In this part we trained a convolutional neural network for image classification using transfer learning. This part involved:

Fine Tuning the convnet: Instead of random initialization, we initialized the network with a pretrained network. Rest of the training was as usual.

ConvNet as fixed feature extractor: Here, we freezed the weights for all of the network except that of the final fully connected layer. This last fully connected layer is replaced with a new one with random weights and only this layer is trained.

Training and Validation:

The training was done on 2275 images of each type and testing was done on 350 images. We then visualized some sample images along with their actual classes. This was followed by the training function of the model. A function for visualizing the data was also written to visualize the images along with the actual class they belonged to. The fully connected convnet was loaded and we reset the final fully connected layer. Then training was done and the maximum accuracy achieved was >90%.