Masked-RCNN for ID Classification and Labeling

Deep Learning - Tensorflow

Omar Farooq 6/25/22

Dataset

The dataset was collected from online resources.

Country	*Number of Original Images*
Pakistan (pk)	32
Singapore (sp)	5
United Kingdom (uk)	5
United States (us)	6

Augmentation was then performed to increase the number of available images. In the augmentation policy:

⇒ The following were always performed:

- Scaling in x and y directions in the range 0.8-1.2
- Translation in x and y directions in the range -0.2% to 0.2%
- Rotation -90 to 90 degrees so any image in any direction may be detected
- Shear in range -8 to 8
- Contrast was adjusted between 0.75 and 1.5
- Multiplication by a number between 0.5 and 1.5

⇒ The following were sometimes performed:

- Adding 3% to 5% gaussian noise
- Blur in the range of 0.5 to 1
- Further translations in range of -0.3% to 0.3% in x and y directions

Four independent models had to be trained for each of the different ID. There for the augmentation policy followed was to equalize the distribution of the various types of IDs.

Country	*Number of Augmented Images*
Pakistan (pk)	352
Singapore (sp)	305
United Kingdom (uk)	305
United States (us)	306

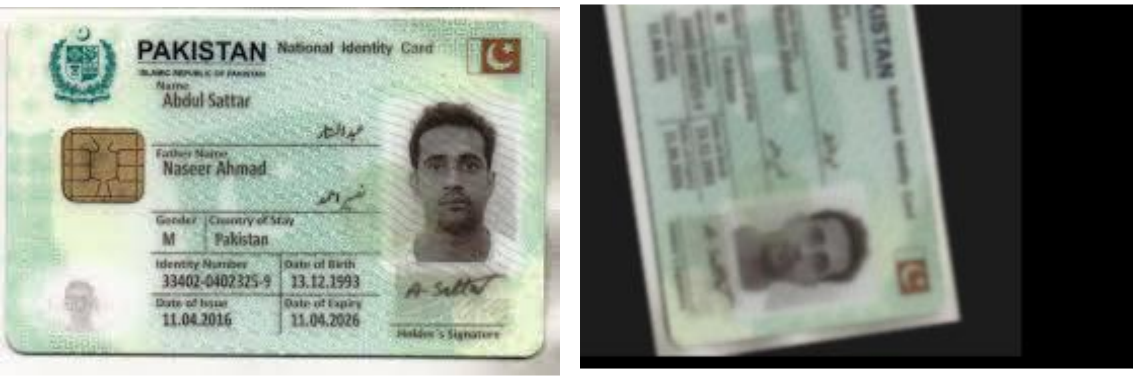
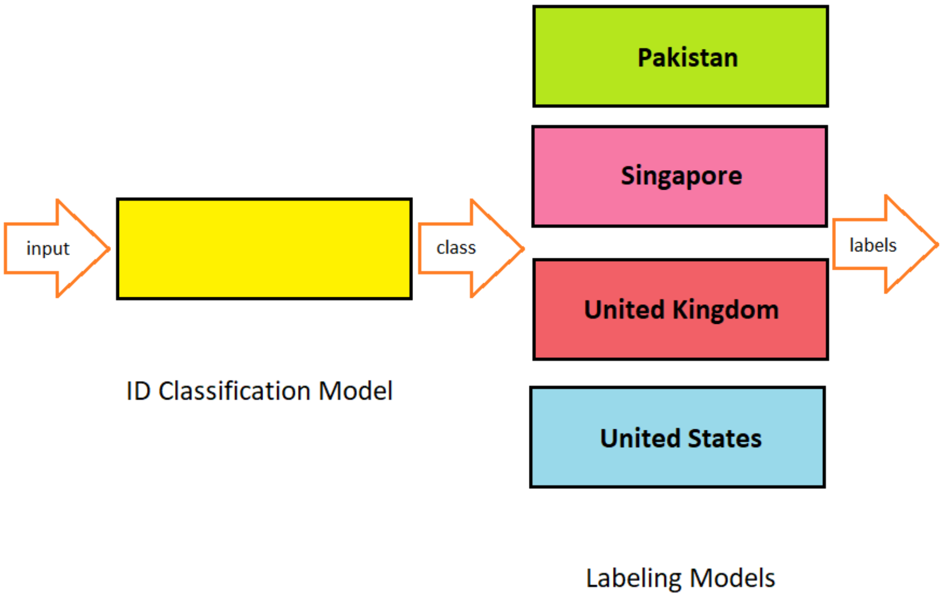
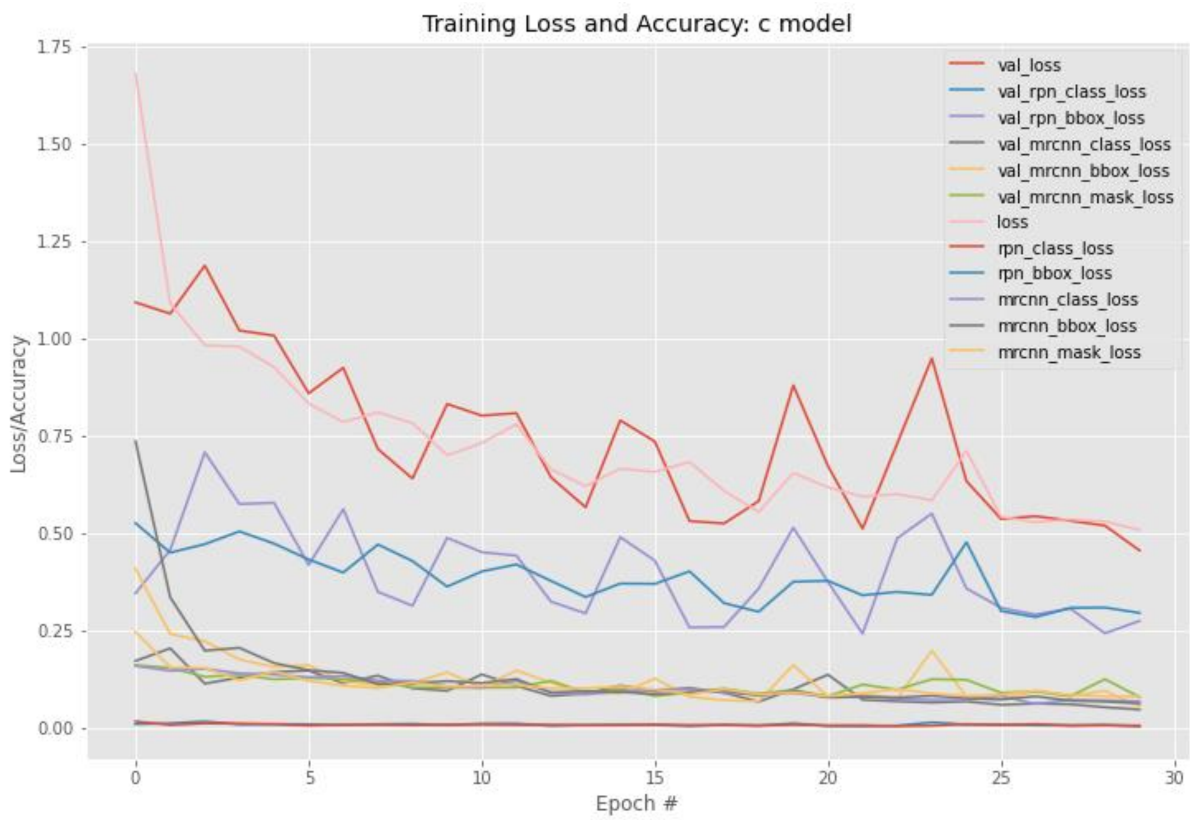


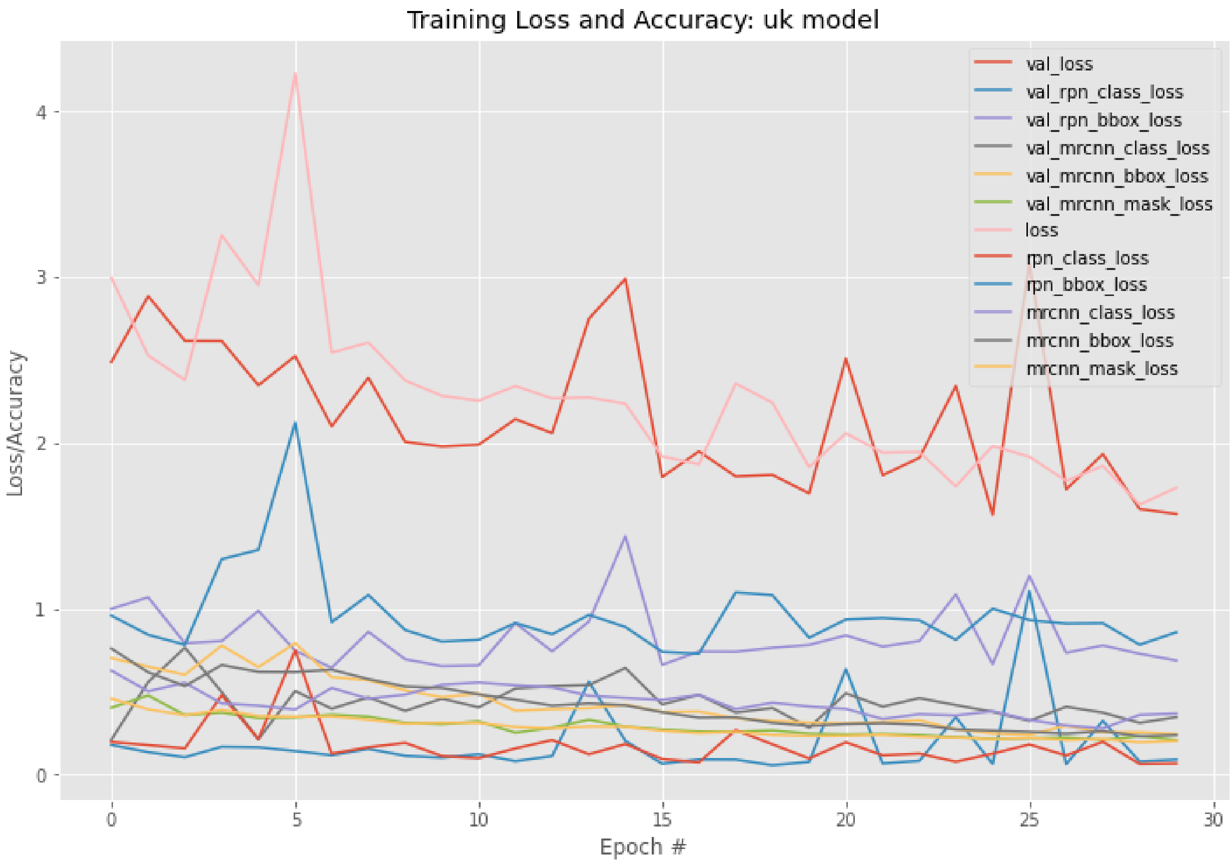
Figure: An example of an original ID image, and its augmented counterpart.

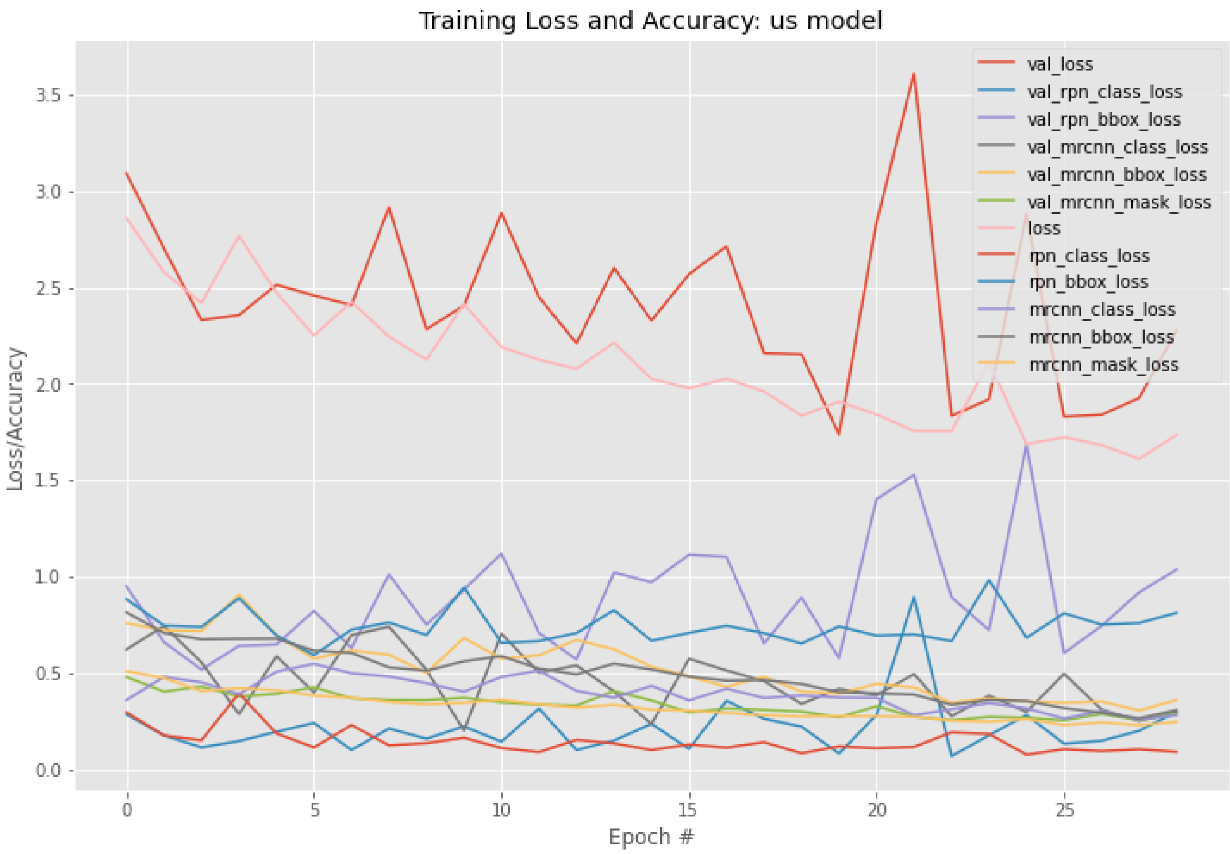
Training

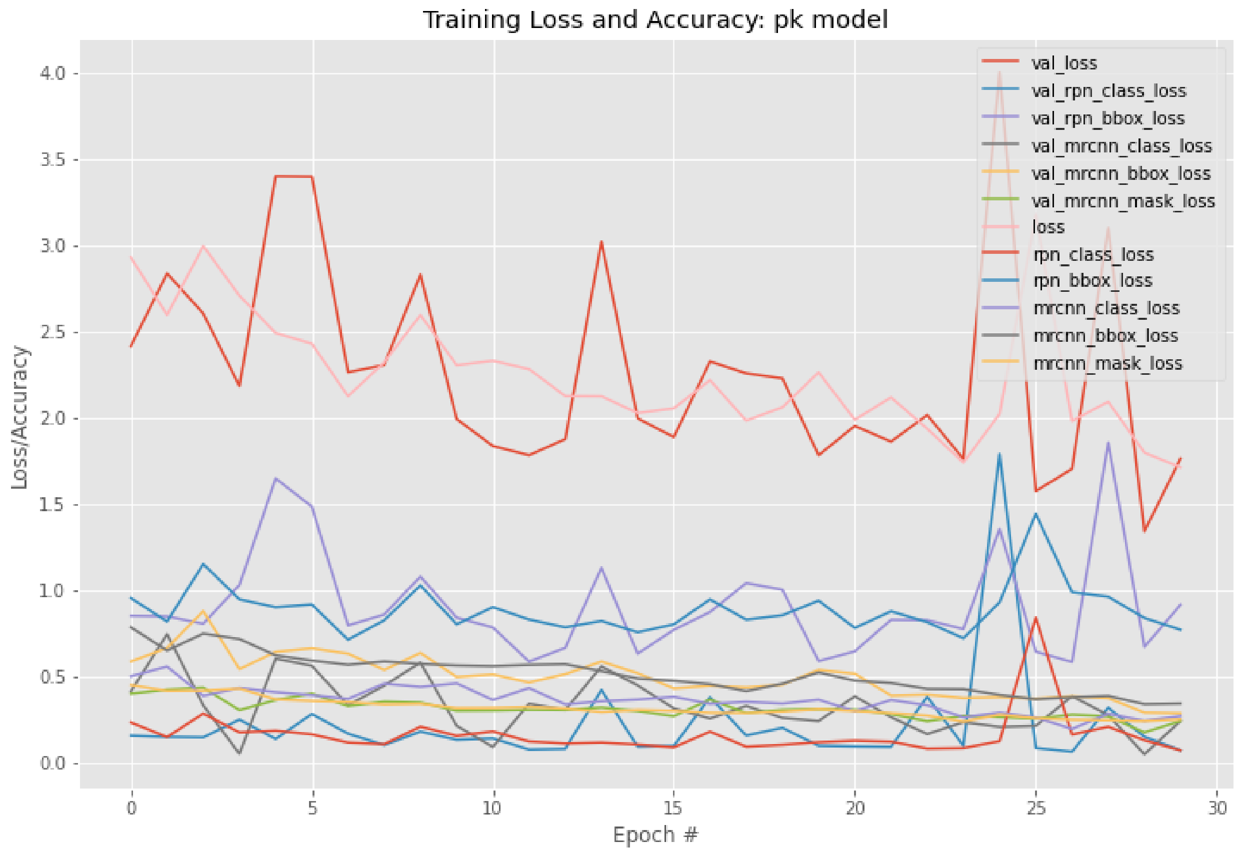
A primary country classification (c) model was trained to classify the images. The images were then fed into their respective labeling model for labeling. Masked RCNN was used for each of these models. A train/test split of 0.8 was used. Pretrained weights from the coco model were used. In this transfer learning scenario, only the head layers were trained. The models were trained for 30 epochs with a learning rate of 0.012. The training and validation losses are plotted for the 5 models on the next few pages.

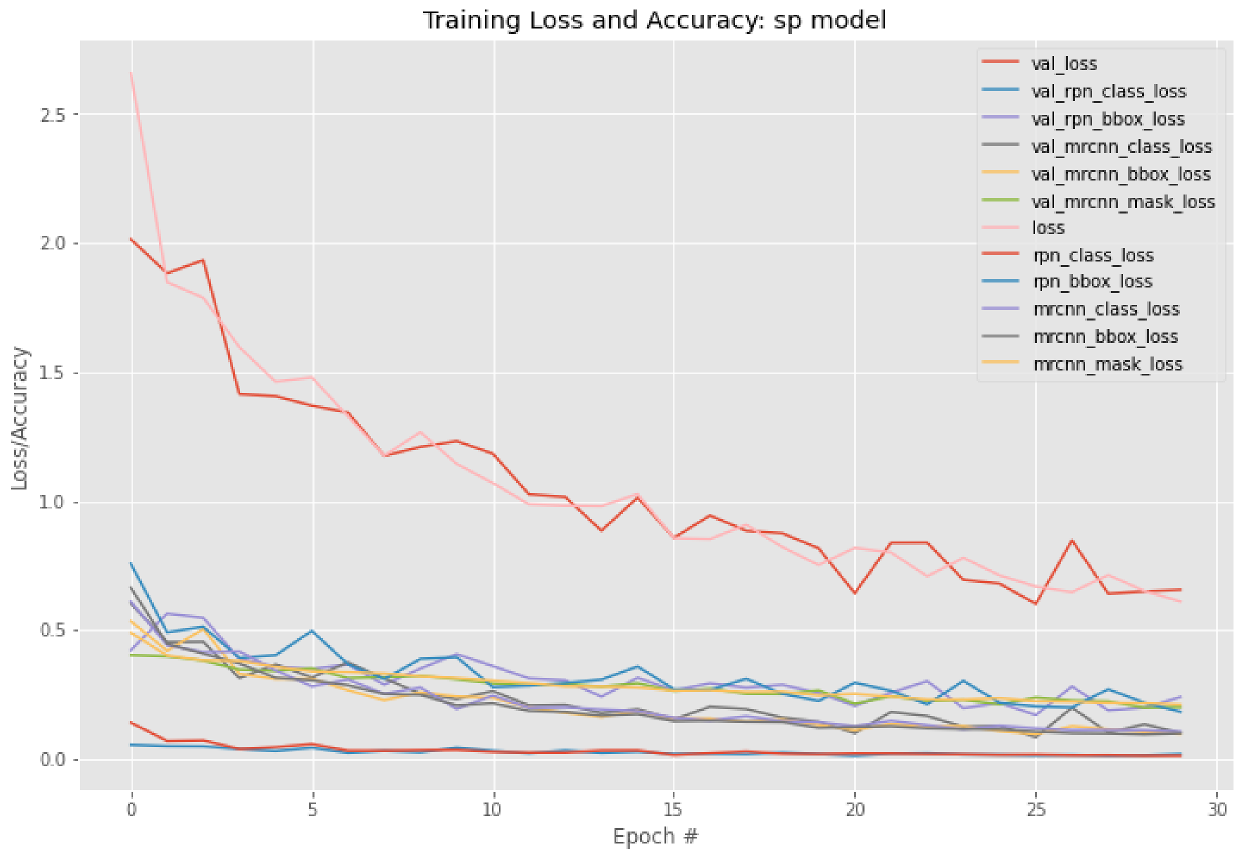





Training Loss and Accuracy: uk model
val_loss
val_rpn_class_loss
val_rpn_bbox_loss
val_mrcnn_class_loss
val_mrcnn_bbox_loss
val_mrcnn_mask_loss
loss
rpn_class_loss
rpn_bbox_loss
mrcnn_class_loss
mrcnn_bbox_loss
mrcnn_mask_loss
Loss/Accuracy
Epoch #
0
1
2
3
4
5
10
15
20
25
30


Training Loss and Accuracy: us model
val_loss
val_rpn_class_loss
val_rpn_bbox_loss
val_mrcnn_class_loss
val_mrcnn_bbox_loss
val_mrcnn_mask_loss
loss
rpn_class_loss
rpn_bbox_loss
mrcnn_class_loss
mrcnn_bbox_loss
mrcnn_mask_loss
Loss/Accuracy
Epoch #
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
0
5
10
15
20
25


Training Loss and Accuracy: pk model
Loss/Accuracy
Epoch #
val_loss
val_rpn_class_loss
val_rpn_bbox_loss
val_mrcnn_class_loss
val_mrcnn_bbox_loss
val_mrcnn_mask_loss
loss
rpn_class_loss
rpn_bbox_loss
mrcnn_class_loss
mrcnn_bbox_loss
mrcnn_mask_loss


Training Loss and Accuracy: sp model
Loss/Accuracy
Epoch #
val_loss
val_rpn_class_loss
val_rpn_bbox_loss
val_mrcnn_class_loss
val_mrcnn_bbox_loss
val_mrcnn_mask_loss
loss
rpn_class_loss
rpn_bbox_loss
mrcnn_class_loss
mrcnn_bbox_loss
mrcnn_mask_loss

The model for labeling Singapore IDs was constantly failing at around the 26th epoch where the gradients would blow up, resulting in a nan loss. The problem was resolved by setting the learning rate to 0.002. The resulting model performed excellently as can be seen in the results portion. Training was performed in a GPU environment on Kaggle. Previous attempts to train the model on Google Colab failed miserably, despite trying all the different versions of the libraries. It took 1.5 minutes per epoch compared to 1 hour per epoch when tried on a laptop without a GPU.

Figure: an example of wrong labeling on the top-left (the entire ID was labeled as gender), compared to the original annotation on top-right. The example on bottom left shows a correctly labeled example.

The model weights were saved when each model was trained. These weights were then loaded into five models. Given any image, the classification model would classify the ID and then the respective model for the predicted country would label it. Intersection-over-Union (IoU) scores were computed for each label and classification. If the IoU score was above 0.4, it was seen as a correctly labeled example. This was then used to compute accuracy of the model.

	PK	SP	UK	US	Final	
Fname	0.51	0.98	0.48	0.66	0.52	**0.63**
Lname	0.46	0.97	0.64	0.56	0.52	**0.63**
DOB	0.27	0.98	0.62	0.55	0.49	**0.58**
gender	0.46	0.93	0.44	0.65	0.53	**0.60**
IDnumber	0.51	0.95	0.49	0.40	0.52	**0.57**
nationality	0.72	1.00	0.44	0.66	0.62	**0.69**
	0.42	**0.97**	**0.44**	**0.5**	**0.46**	**Average**

Table: Accuracy results for labeling

The accuracy for the classification model was computed as 0.75. We see from the above table that the model for the Singapore dataset gave the best accuracy. The Singapore model exceeded even the country classification model despite it having the lowest final loss, and a simple task.