Training Model:

We went with the approach of separate models for Classification and Regression. We tried multiple models like ResNet, Mobilenet, Densenet and are mentioning the finding of the top models according to accuracy on test set.

Classification Modelling:

Problems faced:

* Training and test data size was quite big and with collab, the training of models took a lot of time.

Steps for model training:

* We tried bringing the number of features down (PCA), but the accuracy was less using that (<80), so eventually we did not go ahead with PCA for this use case.
* We used transformations on the train set. We used the FastAI transformation library (<https://fastai1.fast.ai/vision.transform.html>) which is based on the following paper.

https://arxiv.org/abs/1312.5402 - “The techniques include adding more image transformations to training data, adding more transformations to generate additional predictions at test time and using complementary models applied to higher resolution images.“

* We followed the following strategies to train and fine tune our model,
  + Mixup Strategy – Initially we used mixup to remove the memorization and sensitivity problems and once the model was at a decent accuracy, we again trained the model without mixup. This helped us achieve close to 90% Top1 accuracy and above 97% top 5 accuracy, in relatively lesser epochs, which was needed since we were short on training resources (GPU).

“Large deep neural networks are powerful, but exhibit undesirable behaviors such as memorization and sensitivity to adversarial examples. In this work, we propose mixup, a simple learning principle to alleviate these issues. In essence, mixup trains a neural network on convex combinations of pairs of examples and their labels. By doing so, mixup regularizes the neural network to favor simple linear behavior in-between training examples.” – Ref : https://arxiv.org/abs/1710.09412

* + Unfreeze all the layers of the model and training using the learning rate slice argument also experimenting with the weight decay, batch size.
    - Retraining the whole Neural network instead of last layer. max\_lr takes a slice argument. That is telling the model to linearly decrease learning rates as the models goes deeper.
    - Earlier layers will have a smaller learning rate, latter one will have higher. This is due to the fact that as the models goes deeper, each layer learns to capture features that is more specific to the problem domain. We want the early layers to update less as they capture more general features such as edge and corners, which are usually general to many vision problems.
  + Using 1 cycle - Author recommends to do one cycle of learning rate of 2 steps of equal length. We choose maximum learning rate using range test. We use lower learning rate as 1/5th or 1/10th of maximum learning rate. We go from lower learning rate to higher learning rate in step 1 and back to lower learning rate in step 2. We pick this cycle length slightly lesser than total number of epochs to be trained. And in last remaining iterations, we annihilate learning rate way below lower learning rate value(1/10 th or 1/100 th).

The motivation behind this is that, during the middle of learning when learning rate is higher, the learning rate works as regularisation method and keep network from overfitting. This helps the network to avoid steep areas of loss and land better flatter minima. – Ref - https://arxiv.org/pdf/1506.01186.pdf

Training and Testing results:

Resnet152:

Table

Description automatically generated

On Test set:

# loss, accuracy

[0.3675094, tensor(0.9038)]

Top 1 accuracy – 90.34%

Top 5 accuracy – 97.69%

The model got confused between similar looking cars.

Top 5 most confused:

[('Audi S5 Coupe 2012', 'Audi A5 Coupe 2012', 17),

('Audi TTS Coupe 2012', 'Audi TT Hatchback 2011', 16),

('Dodge Caliber Wagon 2012', 'Dodge Caliber Wagon 2007', 14),

('Audi TT Hatchback 2011', 'Audi TTS Coupe 2012', 12),

('BMW 6 Series Convertible 2007', 'BMW M6 Convertible 2010', 12),

Densenet 201

Table

Description automatically generated

# loss, accuracy,

[0.4211438, tensor(0.8939)]

Top 1 accuracy – 89.39%

Top 5 accuracy – 96.8%

The model got confused between similar looking cars.

Top 5 most confused:

[('Audi S5 Coupe 2012', 'Audi A5 Coupe 2012', 16),

('Dodge Caliber Wagon 2012', 'Dodge Caliber Wagon 2007', 14),

('Audi TTS Coupe 2012', 'Audi TT Hatchback 2011', 13),

('Chevrolet Express Cargo Van 2007', 'GMC Savana Van 2012', 13),

('Chevrolet Silverado 1500 Hybrid Crew Cab 2012',

'Chevrolet Silverado 1500 Extended Cab 2012',

13)

**Mobilenet:**

Table

Description automatically generated

loss, accuracy

[0.6534206, tensor(0.8365)]

Top 1 accuracy – 83.6%

Top 5 accuracy – 94.08%

Most confused:

[('Dodge Sprinter Cargo Van 2009', 'Mercedes-Benz Sprinter Van 2012', 13),

('Audi S5 Coupe 2012', 'Audi A5 Coupe 2012', 12),

('BMW 6 Series Convertible 2007', 'BMW M6 Convertible 2010', 12),

('Audi TTS Coupe 2012', 'Audi TT Hatchback 2011', 11),

('Audi V8 Sedan 1994', 'Audi 100 Sedan 1994', 11),

Regression:

We used tensorflow libraries and defined Intersection over Union (IoU) function as an evaluation metrics.

Mobilenet:

A picture containing table

Description automatically generated

Loss, Accuracy

[13.369744300842285, 0.9216721653938293]

Resnet:

Text

Description automatically generated with low confidence

Loss, Accuracy

[17.21445083618164, 0.9168208241462708]

**XceptionNet**

A picture containing table

Description automatically generated

Loss, Accuracy

[14.663599014282227, 0.9170091152191162]

Densenet 121:

Table

Description automatically generated with medium confidence

Loss, Accuracy

[38.32698059082031, 0.8373282551765442]