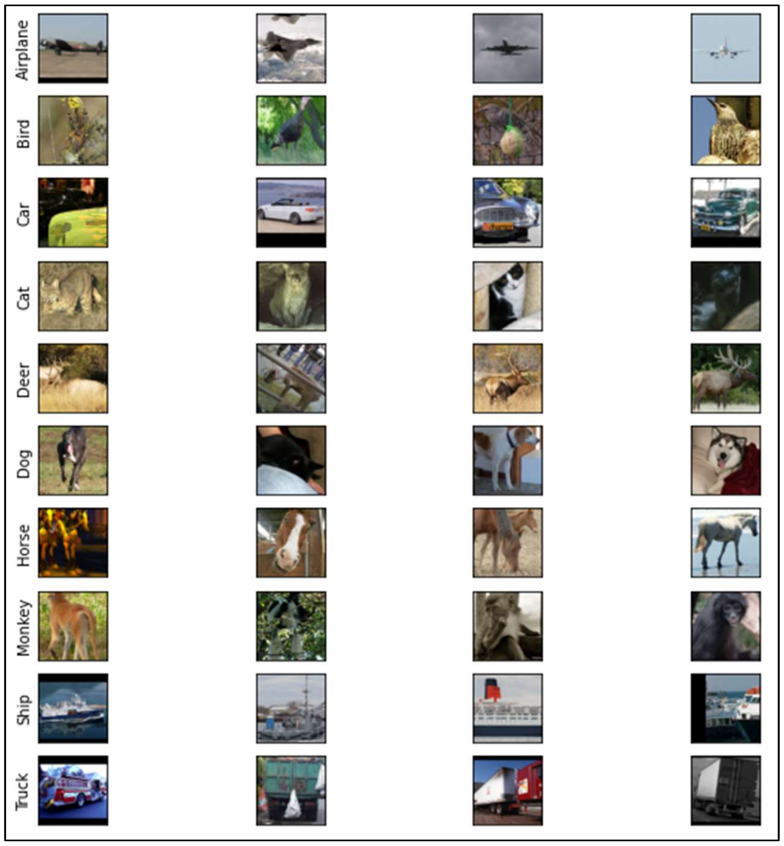
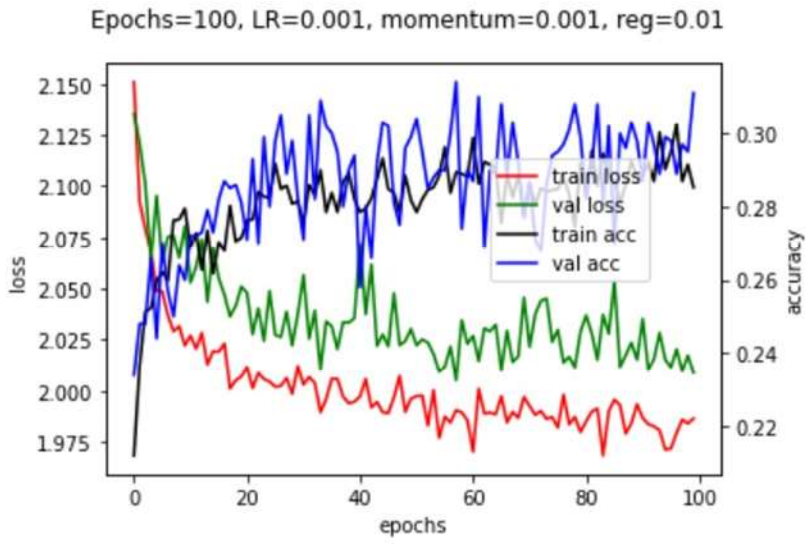
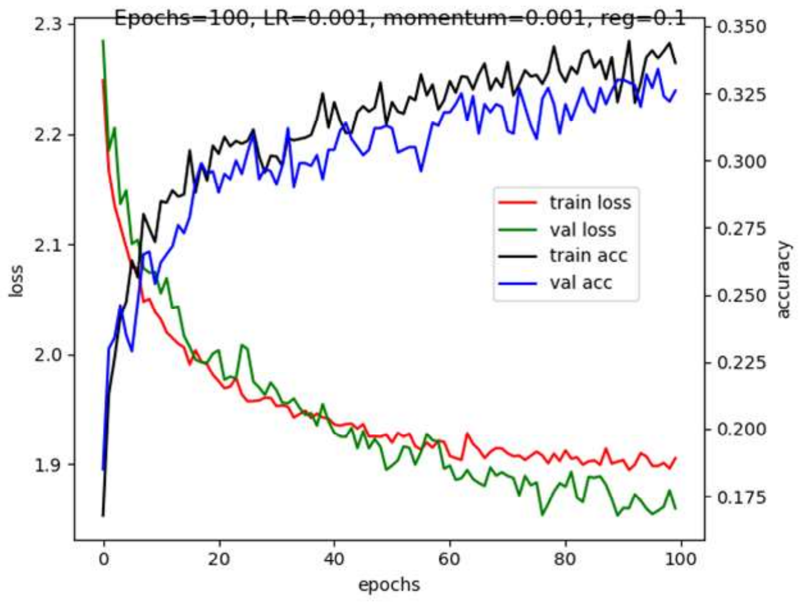
Part 1 - Visualize the Data:



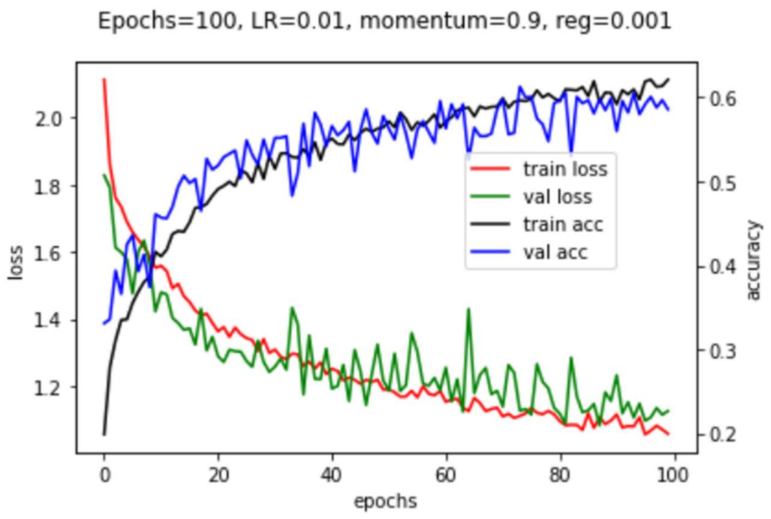
Part 2 – Logistic Regression:



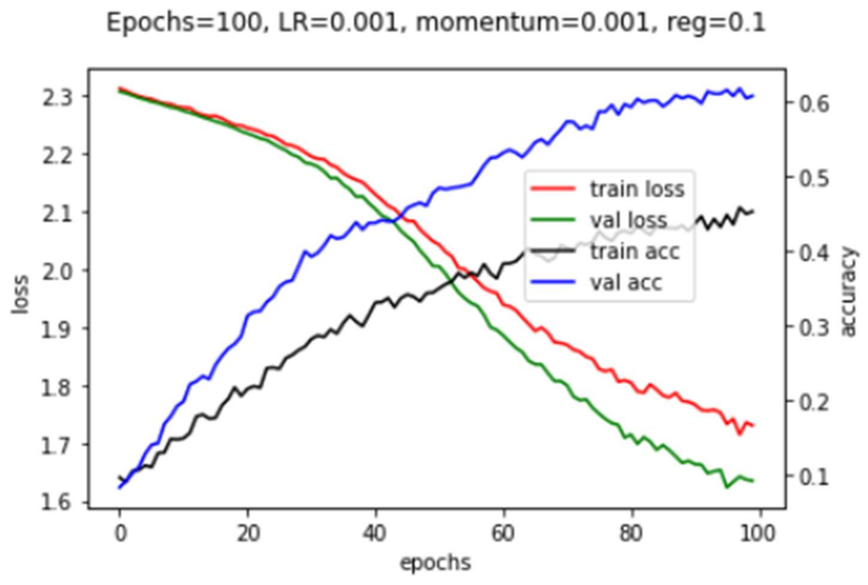
Part 2 – Fully connected NN:



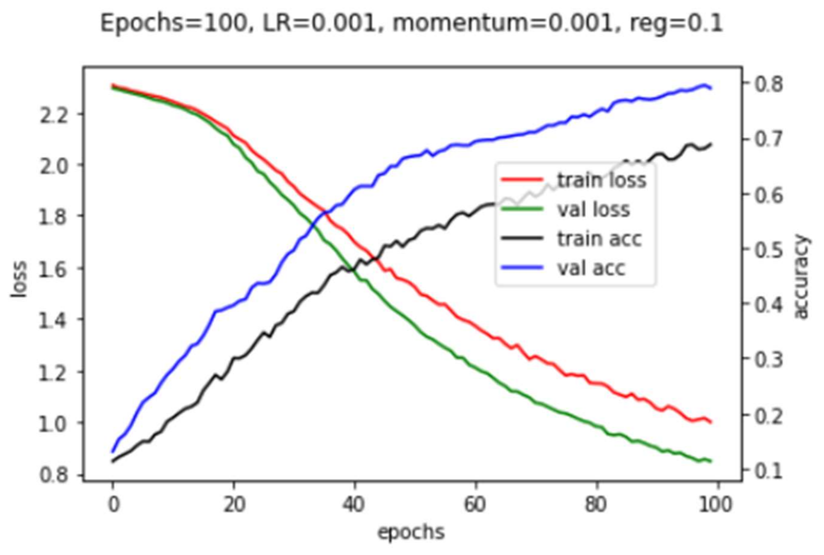
Part 2 – CNN:



Part 2 – Fixed MobileNet-V2:



Part 2 – Learned MobileNet-V2:



Exploring hyper-parameters:

- Batch size:
  We notice that as the batch size is around 64-128 we get the optimal models. In addition, because of the batch normalization if the batch size is very small – less than 50 we get a very bad accuracy.

- Learning rate:
  We notice that as the learning rate decreases the convergence is slower and the loss curve is moderate and stability in its advance. When the learning rate increases the over-fitting increases and it prevents us to find the best parameters, therefore there is big difference between the training accuracy and validation accuracy.
  However, there were no over-fitting problems thanks to the augmentations.

- Regularization coefficient (mom):
  We notice that as the regularization coefficient increases, it decreases the accuracy and the over-fitting.

- Layer size:
  As the layer size increases, the accuracy and runtime increases too, but the model need much more epochs to converge. We can see that as the layer size increases the accuracy and the start point of saturation grows correspondingly.

- Augmentations:
  We notice that too many augmentations did not help the models to learn well, probably since the data changed too much. We used in this dataset only two augmentations, after searching which two would give the best accuracy.

Number of learnable parameters in model:

- NN:
  Linear transformation includes input*output weights + output biases, and batch normalization includes 2*output parameters.
  1. First layer: Input – 64*64*3, output – 200.
     $$64 \cdot 64 \cdot 3 \cdot 200 + 200 + 2 \cdot 200 = 2{,}458{,}200\ parameters$$
  2. Second layer: Input – 200, output – 200.
     $$200 \cdot 200 + 200 + 2 \cdot 200 = 40{,}600\ parameters$$
  3. Third layer: Input – 200, output – 100.
     $$200 \cdot 100 + 100 + 2 \cdot 100 = 20{,}300\ parameters$$
  4. Last layer: Input – 100, output – 10.
     $$100 \cdot 10 + 10 = 1{,}010\ parameters$$

  Total of 2,520,110 parameters.
- CNN:
  Conv layer includes output kernel height*output kernel width*number of output filters*number of input filters + number of output filters, and batch normalization includes 2*output filters. Pooling has no parameters.
  1. First conv layer: filter (3, 3), input (64, 64, 3), output - (31, 31, 6)
     $$3 \cdot 3 \cdot 3 \cdot 6 + 6 + 2 \cdot 6 = 120\ parameters$$
  2. Second conv layer: filter (2, 2), input (31, 31, 6), output - (15, 15, 15)
     $$2 \cdot 2 \cdot 6 \cdot 15 + 15 + 2 \cdot 15 = 405\ parameters$$
  3. Third conv layer: filter (2, 2), input (15, 15, 15), output - (7, 7, 30)
     $$2 \cdot 2 \cdot 15 \cdot 30 + 30 + 2 \cdot 30 = 1890\ parameters$$
  5. First linear layer: Input – 30*7*7, output – 500.
     $$1470 \cdot 500 + 500 = 735{,}500\ parameters$$
  6. Second linear layer: Input – 500, output – 100.
     $$100 \cdot 500 + 100 = 50{,}100\ parameters$$
  7. Last layer: Input – 100, output – 10.
     $$100 \cdot 10 + 10 = 1{,}010\ parameters$$

  Total of 789,025 parameters.