

Object Classification in CIFAR 10

Members:

Praneeth A S, Pratik Satapathy, Jash Dave, Jayant Golhar

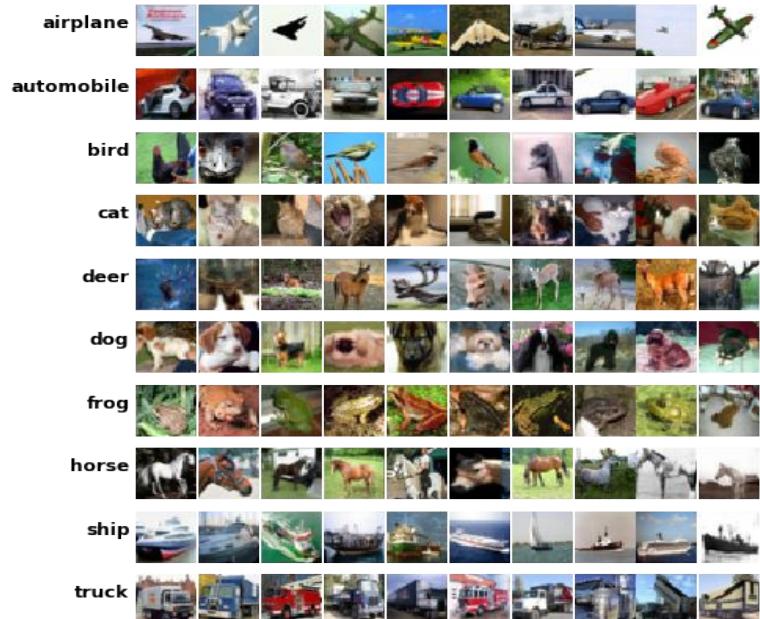


CIFAR 10 Dataset

- Images database consisting of 10 different object classes
- Total 50000 images in the training set
- Total 10000 images in the test set



Train Dataset



Test Dataset

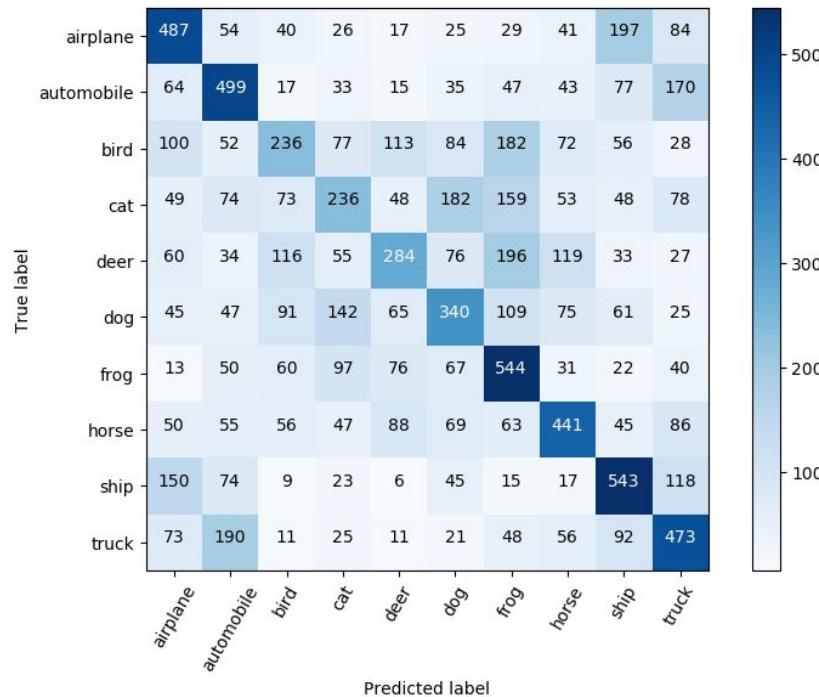
Linear classifier + Raw

Param and accuracy

Test accuracy : 0.4083

- Activation function : relu
- Gradient Descent Method : adam
- Learning rate type : constant
- Learning rate : 0.01
- Regularization constant : 0.01
- Error Tolerance : 1e-07
- Data Transformation method : StandardScaler
- Training epochs : 400

Confusion matrix



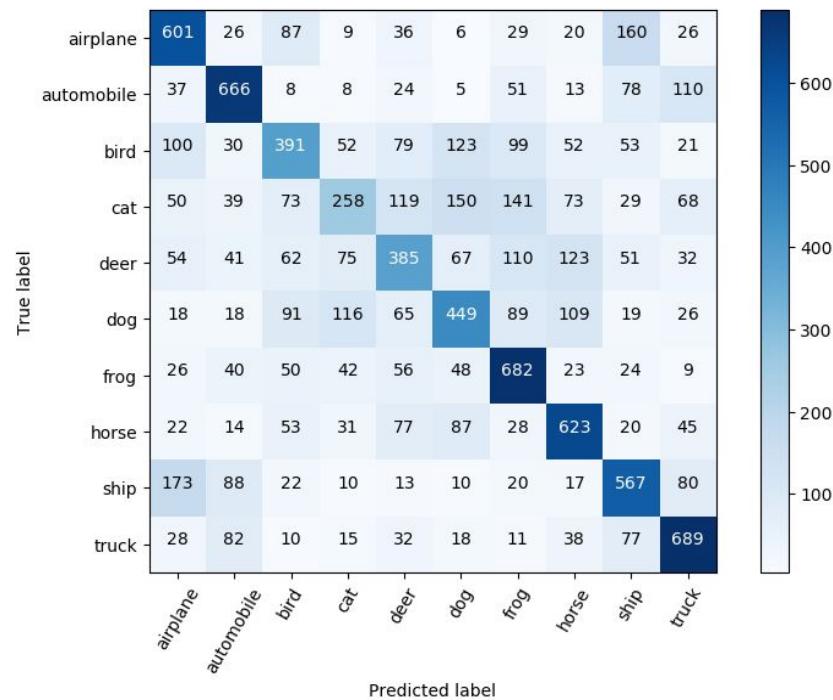
Linear classifier + HOG

Param and accuracy

Test accuracy : 0.5311

- Activation function : relu
- Gradient Descent Method : gradient
- Learning rate type : constant
- Learning rate : 0.01
- Regularization constant : 0.01
- Error Tolerance : 1e-07
- Data Transformation method : StandardScaler
- Training epochs : 1000

Confusion matrix



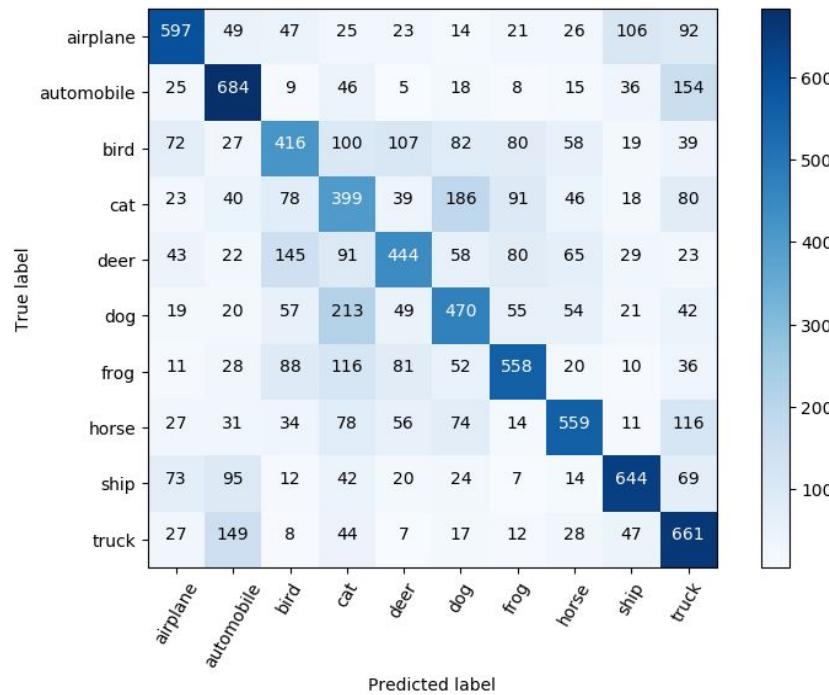
SVM + Raw pixels

Param and accuracy

Test accuracy : 0.5432

- Train validation split : 0.2
- Best parameters
 - svm_kernel = 'rbf'
 - svm_gamma = 0.001
 - svm_c = 100.0
 - svm_max_iter = 10000
 - svm_cs = 1000}
- Accuracy score on test data = 0.5432
- Trained using full train dataset
- Prediction done on 10000 images

Confusion matrix



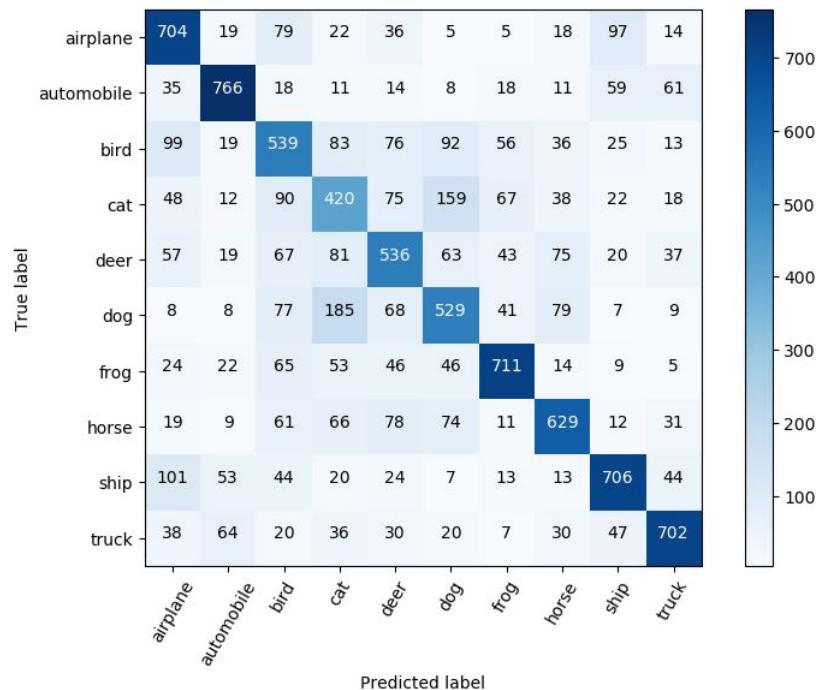
SVM + HOG

Param and accuracy

Test accuracy : 0.6095

- Train Validation split : 0.2
- Best parameters
 - svm_kernel = 'rbf'
 - svm_gamma = 0.001
 - svm_c = 1.0
 - svm_max_iter = 10000
 - svm_cs = 1000
- Accuracy score on test data = 0.6095
- Trained using full train dataset
- Prediction done on 10000 images

Confusion matrix



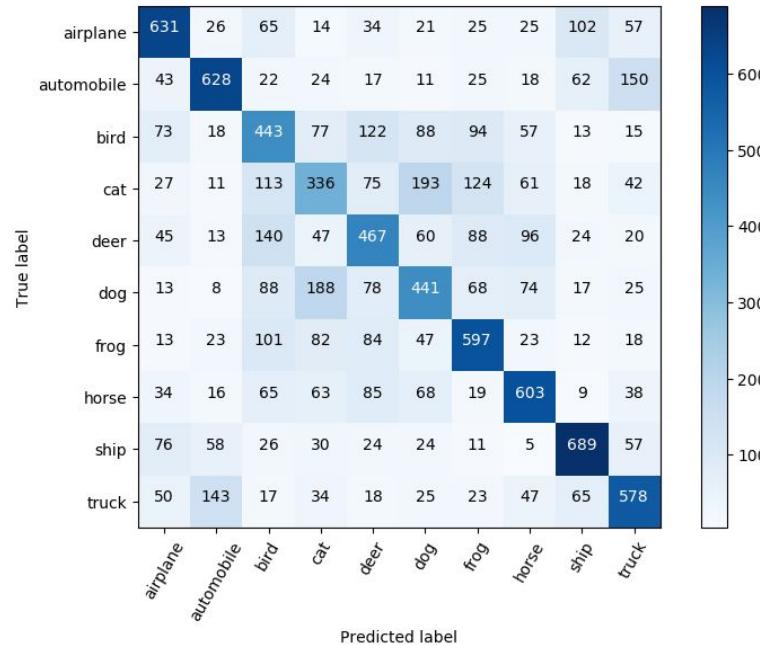
MLP + Raw pixels

Param and accuracy

Test accuracy : 0.5413

- Train test split : 0.2
 - Best parameters mlp_hidden_size = (3072,3072)
 - mlp_iterations = 200
 - mlp_solver = 'adam'
 - Batch_size = 200
- mlp_alpha = 1e-5
- Test accuracy 0.5413
- Trained using full train dataset
- Prediction done on 10000 images

Confusion matrix



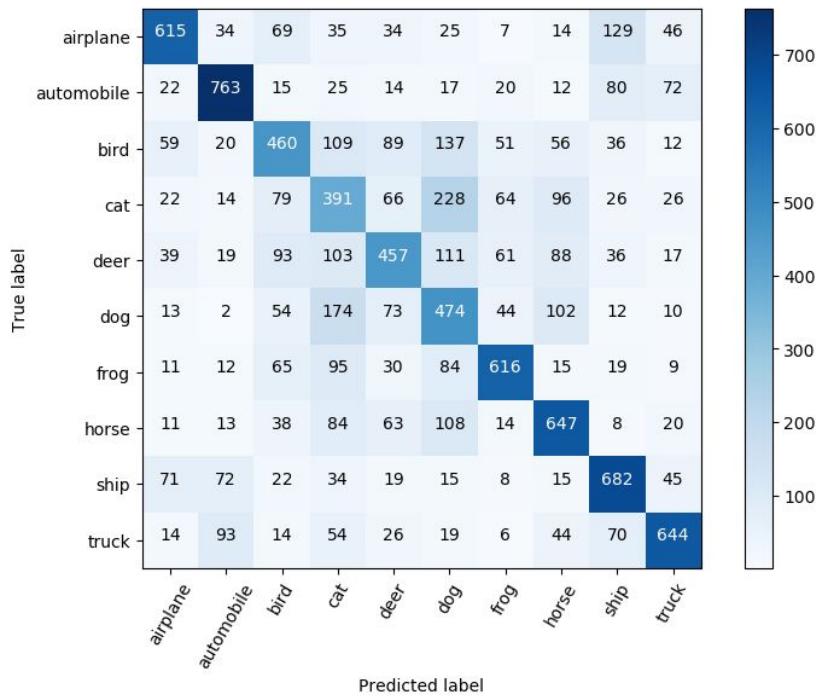
MLP + HOG features

Param and accuracy

Test accuracy : 0.5621

- Train test split : 0.2
- Best parameters
 - mlp_hidden_size = (3072,3072)
 - mlp_iterations = 10000
 - mlp_solver = 'adam'
 - mlp_alpha = 1e-5
 - Batch_size = 200
- Accuracy score on test data = 0.5621
- Trained using full train dataset
- Prediction done on 10000 images

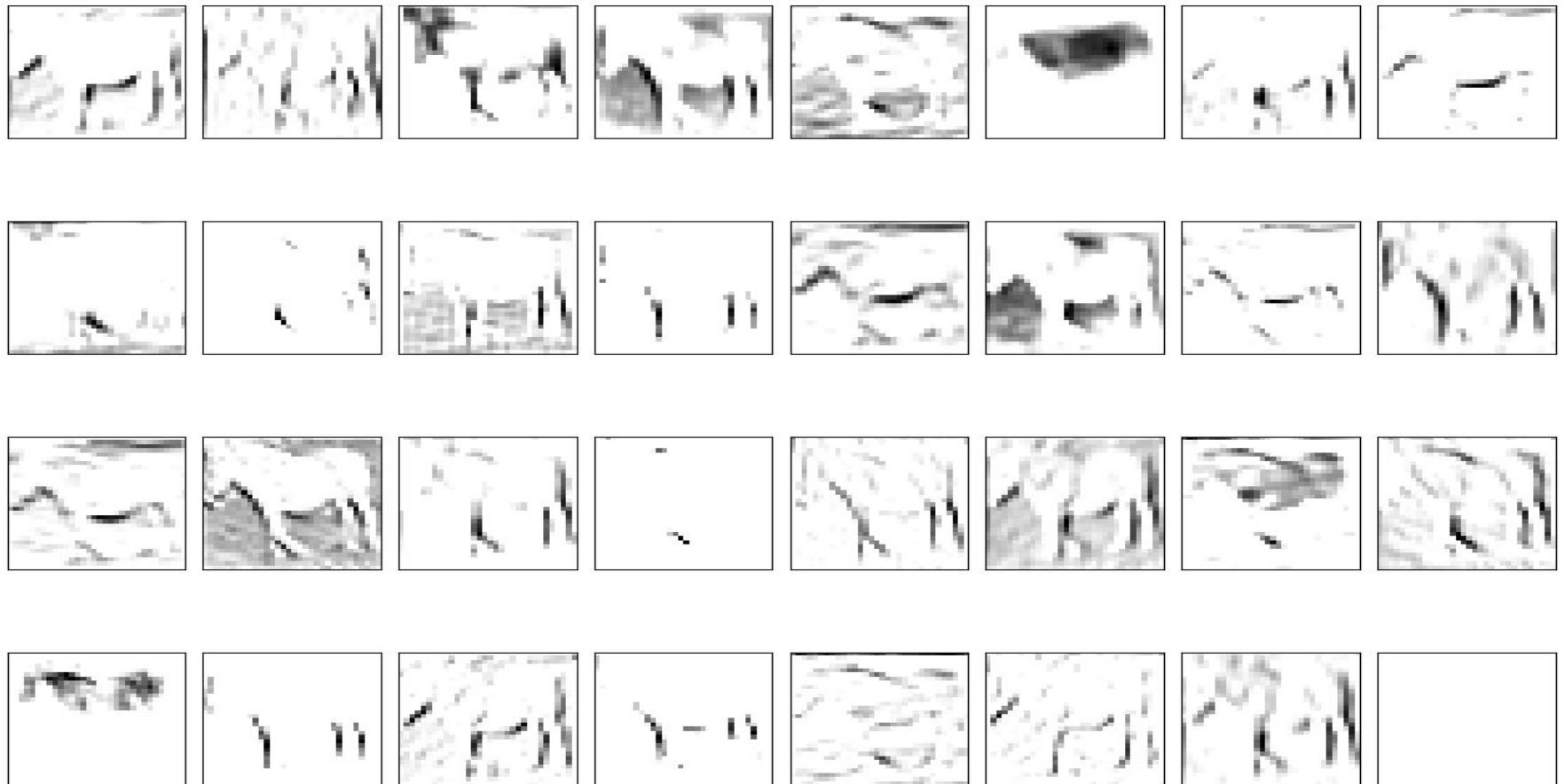
Confusion matrix



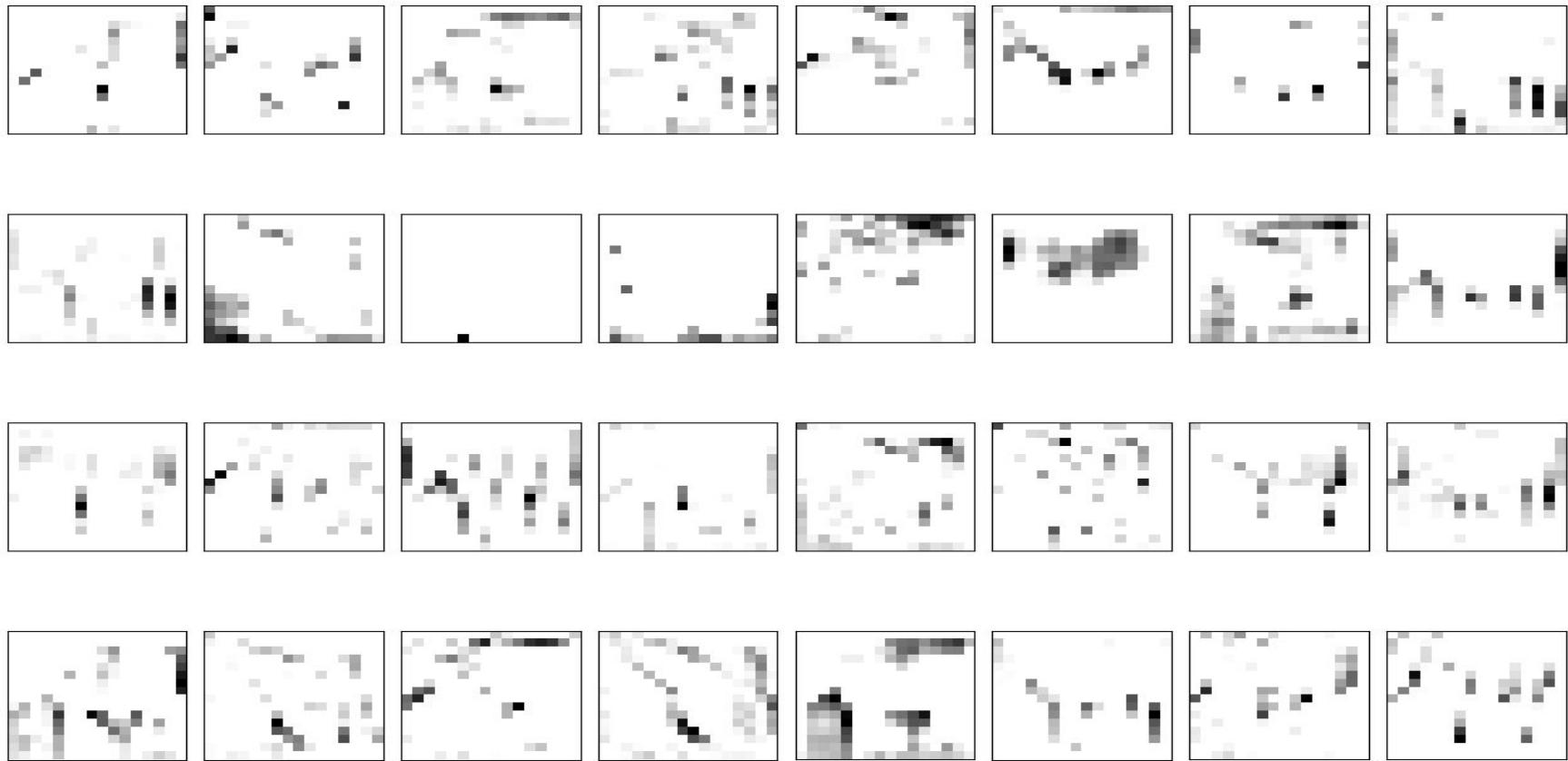
Convolutional Neural Networks

A CNN model with TFLearn

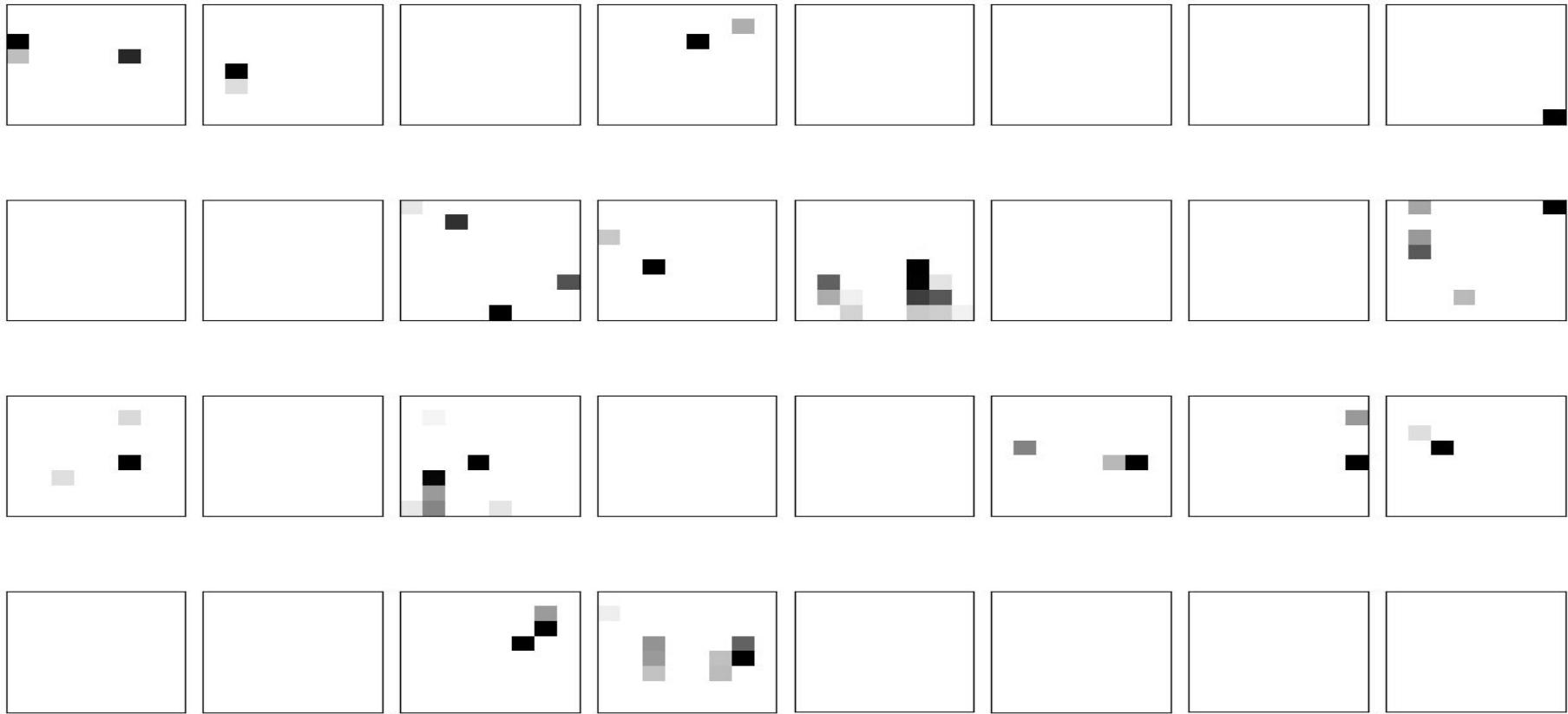
Param and accuracy	Architecture	Confusion matrix																																																																																																																									
<p>Test accuracy : 0.745 (without data augmentation)</p> <p>Test accuracy : 0.786 (with data augmentation)</p> <ul style="list-style-type: none">• Learning_rate=0.001• Activations :<ul style="list-style-type: none">◦ Relu at all layers◦ Softmax at o/p layer• Optimizer='adam'• Batch_size=96• Epoch = 100• Loss=categorical_crossentropy• Feature engineering<ul style="list-style-type: none">◦ Zero center◦ Standard norm◦ Rand flip left right◦ Rand rotation	<p>32 nos 3x3 conv layer Maxpooling layer 2x2</p> <p>64 nos 3x3 conv layer Maxpooling layer 2x2</p> <p>128 nos 3x3 conv layer Maxpooling layer 2x2</p> <p>256 nos 3x3 conv layer Maxpooling layer 2x2</p> <p>512 nos 3x3 conv layer Maxpooling layer 2x2</p> <p>Fully connected layer 1024 Dropout 0.5</p> <p>Fully connected layer 2048 Dropout 0.5</p> <p>Output layer 10</p>	<table border="1"><thead><tr><th>True label \ Predicted label</th><th>airplane</th><th>automobile</th><th>bird</th><th>cat</th><th>deer</th><th>dog</th><th>frog</th><th>horse</th><th>ship</th><th>truck</th></tr></thead><tbody><tr><th>airplane</th><td>785</td><td>17</td><td>30</td><td>12</td><td>16</td><td>4</td><td>5</td><td>10</td><td>105</td><td>16</td></tr><tr><th>automobile</th><td>10</td><td>897</td><td>3</td><td>4</td><td>0</td><td>3</td><td>4</td><td>2</td><td>17</td><td>60</td></tr><tr><th>bird</th><td>49</td><td>8</td><td>702</td><td>44</td><td>72</td><td>26</td><td>47</td><td>37</td><td>11</td><td>4</td></tr><tr><th>cat</th><td>22</td><td>6</td><td>61</td><td>594</td><td>81</td><td>100</td><td>68</td><td>39</td><td>16</td><td>13</td></tr><tr><th>deer</th><td>12</td><td>2</td><td>52</td><td>41</td><td>801</td><td>10</td><td>22</td><td>51</td><td>9</td><td>0</td></tr><tr><th>dog</th><td>14</td><td>6</td><td>38</td><td>173</td><td>46</td><td>644</td><td>18</td><td>48</td><td>9</td><td>4</td></tr><tr><th>frog</th><td>9</td><td>4</td><td>35</td><td>34</td><td>45</td><td>15</td><td>834</td><td>12</td><td>10</td><td>2</td></tr><tr><th>horse</th><td>13</td><td>0</td><td>24</td><td>33</td><td>38</td><td>19</td><td>6</td><td>858</td><td>4</td><td>5</td></tr><tr><th>ship</th><td>26</td><td>16</td><td>9</td><td>13</td><td>1</td><td>1</td><td>5</td><td>1</td><td>917</td><td>11</td></tr><tr><th>truck</th><td>22</td><td>60</td><td>10</td><td>10</td><td>3</td><td>3</td><td>8</td><td>13</td><td>42</td><td>829</td></tr></tbody></table>	True label \ Predicted label	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck	airplane	785	17	30	12	16	4	5	10	105	16	automobile	10	897	3	4	0	3	4	2	17	60	bird	49	8	702	44	72	26	47	37	11	4	cat	22	6	61	594	81	100	68	39	16	13	deer	12	2	52	41	801	10	22	51	9	0	dog	14	6	38	173	46	644	18	48	9	4	frog	9	4	35	34	45	15	834	12	10	2	horse	13	0	24	33	38	19	6	858	4	5	ship	26	16	9	13	1	1	5	1	917	11	truck	22	60	10	10	3	3	8	13	42	829
True label \ Predicted label	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck																																																																																																																	
airplane	785	17	30	12	16	4	5	10	105	16																																																																																																																	
automobile	10	897	3	4	0	3	4	2	17	60																																																																																																																	
bird	49	8	702	44	72	26	47	37	11	4																																																																																																																	
cat	22	6	61	594	81	100	68	39	16	13																																																																																																																	
deer	12	2	52	41	801	10	22	51	9	0																																																																																																																	
dog	14	6	38	173	46	644	18	48	9	4																																																																																																																	
frog	9	4	35	34	45	15	834	12	10	2																																																																																																																	
horse	13	0	24	33	38	19	6	858	4	5																																																																																																																	
ship	26	16	9	13	1	1	5	1	917	11																																																																																																																	
truck	22	60	10	10	3	3	8	13	42	829																																																																																																																	



Convolution layer values for layer 1 for a horse image



Convolution layer values for layer 2 for a horse image



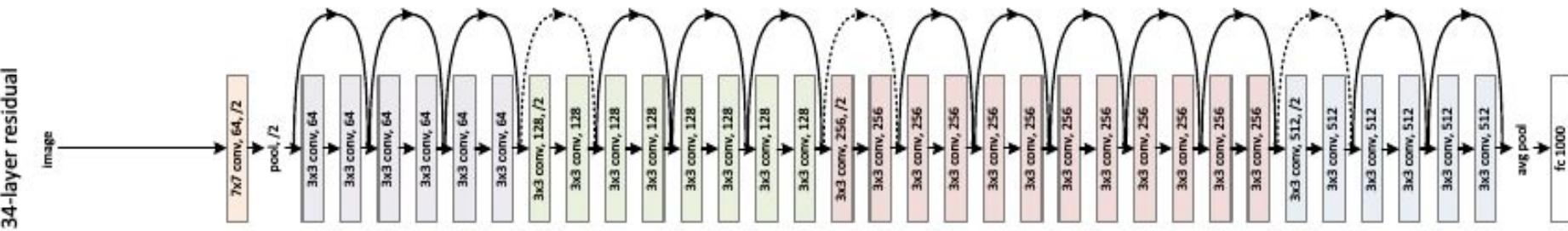
Convolution layer values for layer 3 for horse image

A CNN model (plain non residual design)

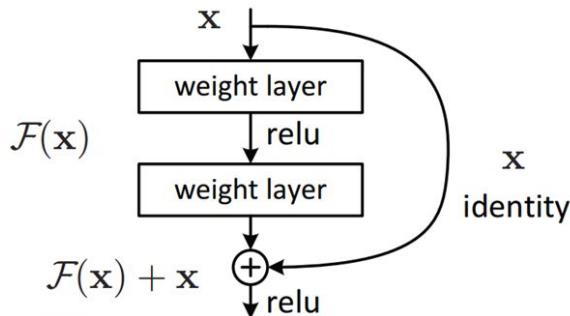
Param and accuracy	Architecture	Confusion matrix																																																																																																																									
<p>Test accuracy : 0.83</p> <ul style="list-style-type: none">• Learning_rate=0.001• Activations :<ul style="list-style-type: none">◦ Relu at all layers◦ Softmax at o/p layer• Optimizer=adam• Loss='categorical_crossentropy'• Batch_size=96• Epoch = 100• Feature engineering<ul style="list-style-type: none">◦ Zero center◦ Standard norm◦ Random flip left right	<p>32 nos 3x3 conv layer (4 layer)</p> <p>Max Pooling layer 2x2</p> <p>64 nos 3x3 conv layer (4 layer)</p> <p>Max Pooling layer 2x2</p> <p>Fully connected layer 512</p> <p>Dropout 0.5</p> <p>Output layer 10</p>	<table border="1"><thead><tr><th>True label \ Predicted label</th><th>airplane</th><th>automobile</th><th>bird</th><th>cat</th><th>deer</th><th>dog</th><th>frog</th><th>horse</th><th>ship</th><th>truck</th></tr></thead><tbody><tr><th>airplane</th><td>853</td><td>11</td><td>42</td><td>17</td><td>5</td><td>2</td><td>7</td><td>10</td><td>40</td><td>13</td></tr><tr><th>automobile</th><td>9</td><td>945</td><td>1</td><td>2</td><td>1</td><td>2</td><td>1</td><td>1</td><td>8</td><td>30</td></tr><tr><th>bird</th><td>43</td><td>4</td><td>779</td><td>48</td><td>23</td><td>39</td><td>42</td><td>15</td><td>1</td><td>6</td></tr><tr><th>cat</th><td>16</td><td>3</td><td>54</td><td>734</td><td>23</td><td>99</td><td>40</td><td>21</td><td>3</td><td>7</td></tr><tr><th>deer</th><td>13</td><td>3</td><td>67</td><td>64</td><td>729</td><td>30</td><td>53</td><td>36</td><td>3</td><td>2</td></tr><tr><th>dog</th><td>10</td><td>4</td><td>31</td><td>113</td><td>12</td><td>788</td><td>13</td><td>26</td><td>1</td><td>2</td></tr><tr><th>frog</th><td>7</td><td>6</td><td>34</td><td>38</td><td>10</td><td>12</td><td>890</td><td>2</td><td>1</td><td>0</td></tr><tr><th>horse</th><td>12</td><td>3</td><td>22</td><td>30</td><td>29</td><td>39</td><td>5</td><td>858</td><td>0</td><td>2</td></tr><tr><th>ship</th><td>59</td><td>19</td><td>12</td><td>11</td><td>4</td><td>3</td><td>4</td><td>4</td><td>871</td><td>13</td></tr><tr><th>truck</th><td>33</td><td>78</td><td>4</td><td>9</td><td>0</td><td>7</td><td>2</td><td>2</td><td>12</td><td>853</td></tr></tbody></table>	True label \ Predicted label	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck	airplane	853	11	42	17	5	2	7	10	40	13	automobile	9	945	1	2	1	2	1	1	8	30	bird	43	4	779	48	23	39	42	15	1	6	cat	16	3	54	734	23	99	40	21	3	7	deer	13	3	67	64	729	30	53	36	3	2	dog	10	4	31	113	12	788	13	26	1	2	frog	7	6	34	38	10	12	890	2	1	0	horse	12	3	22	30	29	39	5	858	0	2	ship	59	19	12	11	4	3	4	4	871	13	truck	33	78	4	9	0	7	2	2	12	853
True label \ Predicted label	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck																																																																																																																	
airplane	853	11	42	17	5	2	7	10	40	13																																																																																																																	
automobile	9	945	1	2	1	2	1	1	8	30																																																																																																																	
bird	43	4	779	48	23	39	42	15	1	6																																																																																																																	
cat	16	3	54	734	23	99	40	21	3	7																																																																																																																	
deer	13	3	67	64	729	30	53	36	3	2																																																																																																																	
dog	10	4	31	113	12	788	13	26	1	2																																																																																																																	
frog	7	6	34	38	10	12	890	2	1	0																																																																																																																	
horse	12	3	22	30	29	39	5	858	0	2																																																																																																																	
ship	59	19	12	11	4	3	4	4	871	13																																																																																																																	
truck	33	78	4	9	0	7	2	2	12	853																																																																																																																	

Microsoft ResNet (winner of ILSVRC 2015)

Resnet architecture :



Residual block
architecture :

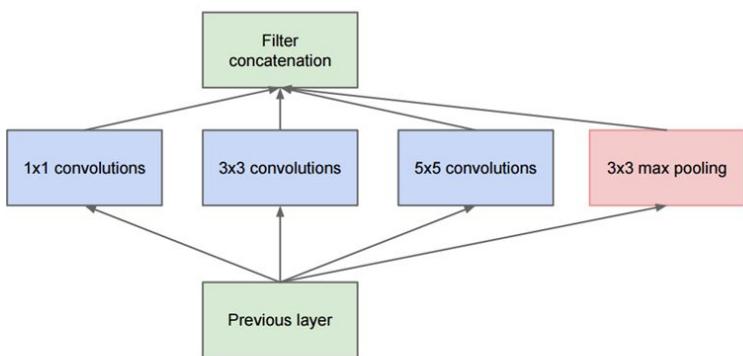
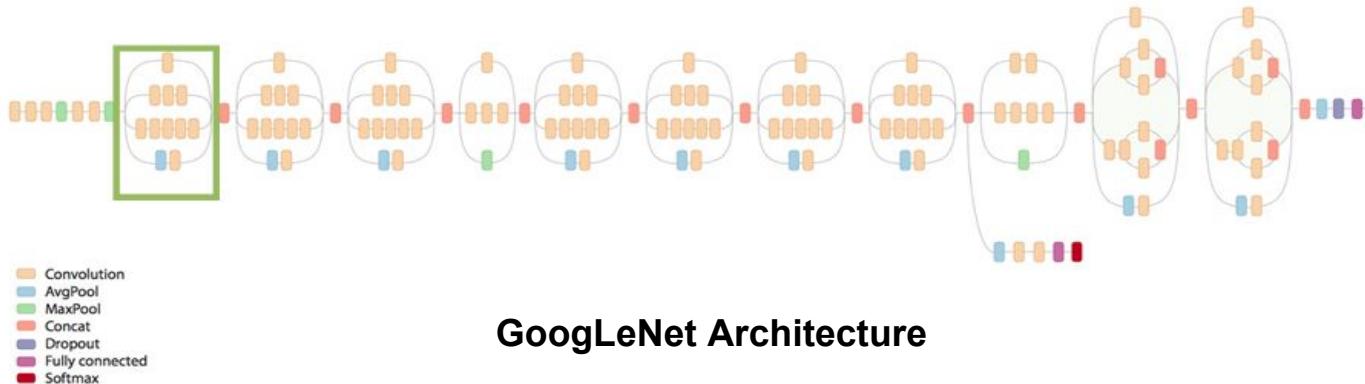


Ref: <https://arxiv.org/pdf/1512.03385.pdf>

A smaller CNN inspired from Microsoft (ResNet)

Param and accuracy	Architecture	Confusion matrix																																																																																																																									
<p>Test accuracy : 0.87</p> <ul style="list-style-type: none">• Learning_rate=0.001• Activations :<ul style="list-style-type: none">◦ Relu at all layers◦ Softmax at o/p layer• Optimizer=momentum• Loss=categorical_crossentropy• Batch_size=128• Epoch = 100• Feature engineering<ul style="list-style-type: none">◦ Zero center◦ Standard norm◦ Ran flip left right◦ Rand rotation◦ Rand rotation/crop	<p>16 nos 3x3 conv layer</p> <p>5 residual block 16 o/p ch</p> <p>5 residual block 32 o/p ch</p> <p>5 residual block 64 o/p ch</p> <p>Relu activation</p> <p>Global avg pool</p> <p>Fully connected o/p layer 10</p>	<table border="1"><thead><tr><th>True label \ Predicted label</th><th>airplane</th><th>automobile</th><th>bird</th><th>cat</th><th>deer</th><th>dog</th><th>frog</th><th>horse</th><th>ship</th><th>truck</th></tr></thead><tbody><tr><th>airplane</th><td>915</td><td>13</td><td>28</td><td>4</td><td>7</td><td>3</td><td>4</td><td>4</td><td>18</td><td>4</td></tr><tr><th>automobile</th><td>13</td><td>960</td><td>2</td><td>0</td><td>0</td><td>1</td><td>0</td><td>2</td><td>4</td><td>18</td></tr><tr><th>bird</th><td>20</td><td>0</td><td>919</td><td>13</td><td>15</td><td>17</td><td>3</td><td>8</td><td>4</td><td>1</td></tr><tr><th>cat</th><td>14</td><td>2</td><td>92</td><td>716</td><td>29</td><td>122</td><td>5</td><td>12</td><td>6</td><td>2</td></tr><tr><th>deer</th><td>4</td><td>1</td><td>79</td><td>10</td><td>845</td><td>24</td><td>5</td><td>27</td><td>5</td><td>0</td></tr><tr><th>dog</th><td>3</td><td>1</td><td>55</td><td>54</td><td>22</td><td>846</td><td>1</td><td>12</td><td>4</td><td>2</td></tr><tr><th>frog</th><td>1</td><td>2</td><td>147</td><td>50</td><td>18</td><td>21</td><td>756</td><td>2</td><td>2</td><td>1</td></tr><tr><th>horse</th><td>2</td><td>1</td><td>32</td><td>13</td><td>12</td><td>29</td><td>1</td><td>904</td><td>5</td><td>1</td></tr><tr><th>ship</th><td>29</td><td>19</td><td>11</td><td>0</td><td>1</td><td>2</td><td>0</td><td>0</td><td>932</td><td>6</td></tr><tr><th>truck</th><td>34</td><td>53</td><td>6</td><td>6</td><td>0</td><td>0</td><td>0</td><td>3</td><td>18</td><td>880</td></tr></tbody></table>	True label \ Predicted label	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck	airplane	915	13	28	4	7	3	4	4	18	4	automobile	13	960	2	0	0	1	0	2	4	18	bird	20	0	919	13	15	17	3	8	4	1	cat	14	2	92	716	29	122	5	12	6	2	deer	4	1	79	10	845	24	5	27	5	0	dog	3	1	55	54	22	846	1	12	4	2	frog	1	2	147	50	18	21	756	2	2	1	horse	2	1	32	13	12	29	1	904	5	1	ship	29	19	11	0	1	2	0	0	932	6	truck	34	53	6	6	0	0	0	3	18	880
True label \ Predicted label	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck																																																																																																																	
airplane	915	13	28	4	7	3	4	4	18	4																																																																																																																	
automobile	13	960	2	0	0	1	0	2	4	18																																																																																																																	
bird	20	0	919	13	15	17	3	8	4	1																																																																																																																	
cat	14	2	92	716	29	122	5	12	6	2																																																																																																																	
deer	4	1	79	10	845	24	5	27	5	0																																																																																																																	
dog	3	1	55	54	22	846	1	12	4	2																																																																																																																	
frog	1	2	147	50	18	21	756	2	2	1																																																																																																																	
horse	2	1	32	13	12	29	1	904	5	1																																																																																																																	
ship	29	19	11	0	1	2	0	0	932	6																																																																																																																	
truck	34	53	6	6	0	0	0	3	18	880																																																																																																																	

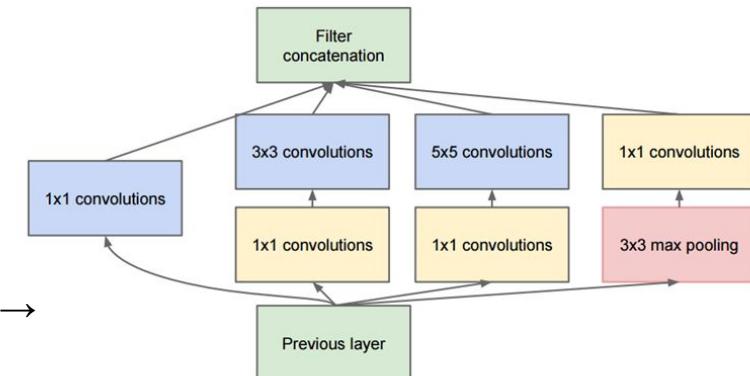
CNN transfer learning with Inception model



← Building blocks →

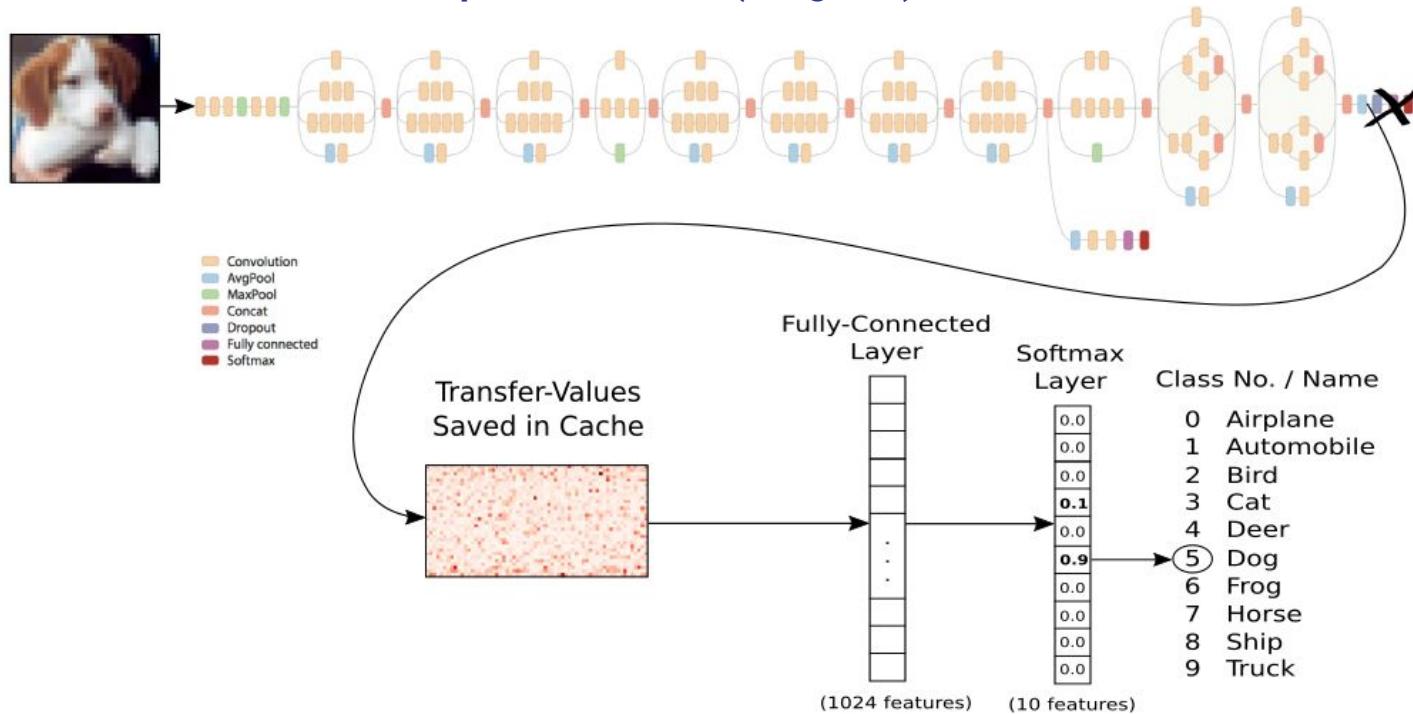
Naive idea of an Inception module

Full Inception module



Transfer Learning

Inception model - v3(imagenet) to CIFAR10



Inception v3 transfer learning to CIFAR10

Param and accuracy	Architecture	Confusion matrix																																																																																																																									
<p>Test accuracy : 0.9010</p> <ul style="list-style-type: none">• Learning_rate=0.0001• Activations :<ul style="list-style-type: none">◦ Relu at all layers◦ Softmax at o/p layer• Optimizer=adam• Loss='categorical_crossentropy'• Batch_size=256• Epoch = 10000	<p>Inception model (non trainable layers)</p> <p>.</p> <p>.</p> <p>.</p> <p>Fully connected layer 100 (trainable layer)</p> <p>Fully connected layer 10 (trainable layer)</p> <p>Softmax classifier</p>	<p>True label</p> <table border="1"><thead><tr><th></th><th>airplane</th><th>automobile</th><th>bird</th><th>cat</th><th>deer</th><th>dog</th><th>frog</th><th>horse</th><th>ship</th><th>truck</th></tr></thead><tbody><tr><th>airplane</th><td>902</td><td>10</td><td>16</td><td>5</td><td>7</td><td>0</td><td>3</td><td>3</td><td>38</td><td>16</td></tr><tr><th>automobile</th><td>8</td><td>950</td><td>3</td><td>2</td><td>0</td><td>1</td><td>2</td><td>1</td><td>5</td><td>28</td></tr><tr><th>bird</th><td>16</td><td>1</td><td>891</td><td>20</td><td>27</td><td>13</td><td>20</td><td>8</td><td>3</td><td>1</td></tr><tr><th>cat</th><td>9</td><td>4</td><td>16</td><td>770</td><td>20</td><td>126</td><td>31</td><td>18</td><td>4</td><td>2</td></tr><tr><th>deer</th><td>4</td><td>0</td><td>36</td><td>17</td><td>867</td><td>19</td><td>21</td><td>33</td><td>2</td><td>1</td></tr><tr><th>dog</th><td>3</td><td>0</td><td>12</td><td>42</td><td>13</td><td>906</td><td>7</td><td>15</td><td>1</td><td>1</td></tr><tr><th>frog</th><td>2</td><td>1</td><td>22</td><td>12</td><td>18</td><td>9</td><td>933</td><td>1</td><td>2</td><td>0</td></tr><tr><th>horse</th><td>1</td><td>0</td><td>7</td><td>10</td><td>22</td><td>29</td><td>2</td><td>919</td><td>7</td><td>3</td></tr><tr><th>ship</th><td>36</td><td>11</td><td>0</td><td>3</td><td>0</td><td>0</td><td>3</td><td>1</td><td>938</td><td>8</td></tr><tr><th>truck</th><td>8</td><td>38</td><td>2</td><td>0</td><td>0</td><td>1</td><td>1</td><td>2</td><td>9</td><td>939</td></tr></tbody></table> <p>Predicted label</p>		airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck	airplane	902	10	16	5	7	0	3	3	38	16	automobile	8	950	3	2	0	1	2	1	5	28	bird	16	1	891	20	27	13	20	8	3	1	cat	9	4	16	770	20	126	31	18	4	2	deer	4	0	36	17	867	19	21	33	2	1	dog	3	0	12	42	13	906	7	15	1	1	frog	2	1	22	12	18	9	933	1	2	0	horse	1	0	7	10	22	29	2	919	7	3	ship	36	11	0	3	0	0	3	1	938	8	truck	8	38	2	0	0	1	1	2	9	939
	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck																																																																																																																	
airplane	902	10	16	5	7	0	3	3	38	16																																																																																																																	
automobile	8	950	3	2	0	1	2	1	5	28																																																																																																																	
bird	16	1	891	20	27	13	20	8	3	1																																																																																																																	
cat	9	4	16	770	20	126	31	18	4	2																																																																																																																	
deer	4	0	36	17	867	19	21	33	2	1																																																																																																																	
dog	3	0	12	42	13	906	7	15	1	1																																																																																																																	
frog	2	1	22	12	18	9	933	1	2	0																																																																																																																	
horse	1	0	7	10	22	29	2	919	7	3																																																																																																																	
ship	36	11	0	3	0	0	3	1	938	8																																																																																																																	
truck	8	38	2	0	0	1	1	2	9	939																																																																																																																	

Summary of Experiments

Model#	Approach + Input data	Parameters (Activation function, training epochs, learning rate)	Training Accuracy	Validation Accuracy	Test Accuracy
1	Linear Classifier + Raw	Softmax, 400, 0.01	0.4243	0.4156	0.4083
2	Linear Classifier + HOG	Softmax, 1000, 0.01	0.5494	0.5324	0.5311
3	SVM + Raw	RBF, gamma=0.001, c=100, cs=1000	-	0.5353	0.5432
4	SVM + HOG	RBF, gamma=0.001, c=1, cs=1000	-	0.6127	0.6193
5	MLP + Raw	Sigmoid, 1000, 1.5, hidden: (3072, 3072)	1.0	0.5448	0.5413
6	MLP + HOG	alpha:1e-05, hidden: (3072, 3072), adam	-	0.5749	0.5621

Summary of Experiments (contd..)

Model#	Approach + Input data	Parameters (Activation function, training epochs, learning rate)	Training Accuracy	Validation Accuracy	Test Accuracy
7	CNN	Relu, 100, 0.001, optimizer: adam [model arch in slide : 12]	0.8926	0.7915	0.7861
8	CNN non residual	Relu, 100, 0.001, optimizer: adam [model arch in slide : 13]	0.9072	0.8362	0.8299
9	CNN Resnet inspired	Relu, 100, 0.001, optimizer: momentum [model arch in slide : 15]	0.9471	0.8684	0.8673
10	Transfer learning (Inception)	Relu, 10000, 0.0001, optimizer: adam [model arch in slide : 14]	0.9801	-	0.9015

References

1. Navneet Dalal, Bill Triggs - Histogram of Oriented Gradients - CVPR 2005
2. Corinna Cortes, Vladimir Vapnik - Support Vector Networks - Machine Learning Journal
3. CIFAR-10 - <https://www.cs.toronto.edu/~kriz/cifar.html>, last visited 27/04/2017
4. Tutorials for Inception model - <https://github.com/Hvass-Labs/TensorFlow-Tutorials>, last visited 27/04/2017
5. Inception paper : <https://arxiv.org/pdf/1409.4842v1.pdf>
6. <http://tflearn.org/>
7. <https://www.tensorflow.org/>
8. Resnet <https://arxiv.org/pdf/1512.03385.pdf>
9. <https://github.com/tflearn/tflearn> repository
10. Alexnet <https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>