Name: SHAH, MIT KALPESHKUMAR **UT EID**: mks3226
Visual Recognition *Coding Assignment 2- Recognizing scene categories*

Implementation:

AlexNet implementation from Caffe was used.

In particular, I used it for 4 different cases:

Case A: **No** pre-training, **No** extra-images

Case B: **With** pre-training, **No** extra-images

Case C: **No** pre-training, **With** extra-images

Case D: **With** pre-training, **With** extra-images

Pre-processing

- First of all, training.txt & testing.txt files were generated from the given 'filenames.mat'.
- Then they were converted to "lmdb format" by running the script /caffe/examples/imagenet/create_imagenet.sh
- After that 'scene_mean.binaryproto' file (mean image file) was created using /caffe/examples/imagenet/make_imagenet_mean.sh
- Then, train_val.protxt & solver.protxt were copied from /caffe/models/bvlc_Alexnet and path changes were made accordingly.
- fc8 in train_val.protxt was modified to be of 25 instead of 1000 for the current task.
- Specific changes in parameters (mainly in solver.protxt) are mentioned for each of the cases separately.

Details for each of the case are as follows:

<u>Case A:</u> <u>No</u> pre-training, **<u>No</u>** extra-images

Only 2500 training images mentioned were used for it & was tested on 625 images.

<u>Training</u>

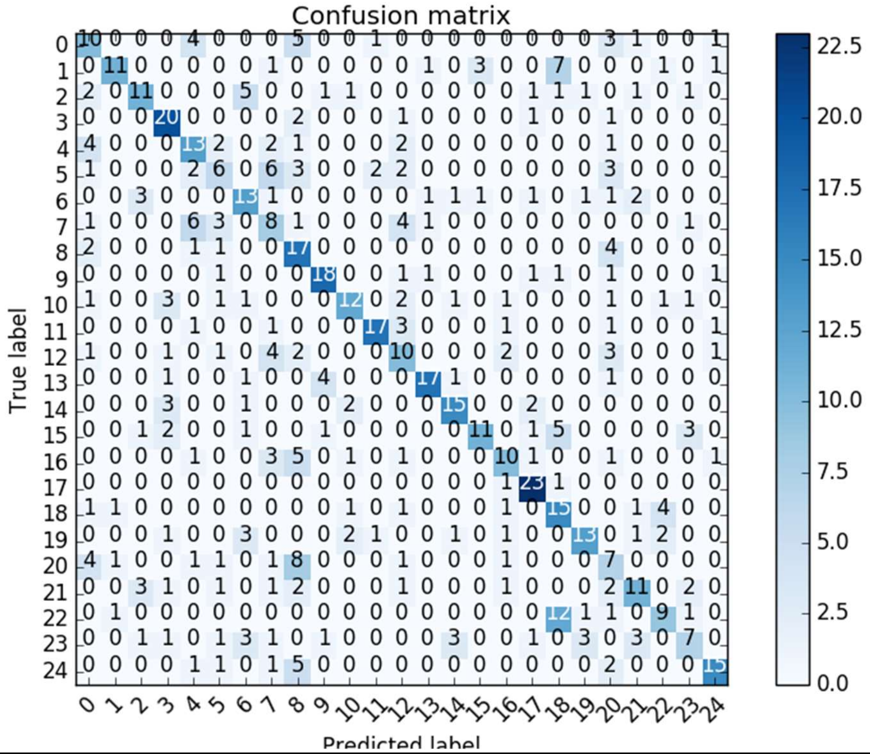
- Network was initialized **randomly**
- Training was done for **two learning rates** – 0.01, 0.001

<u>Logs</u>

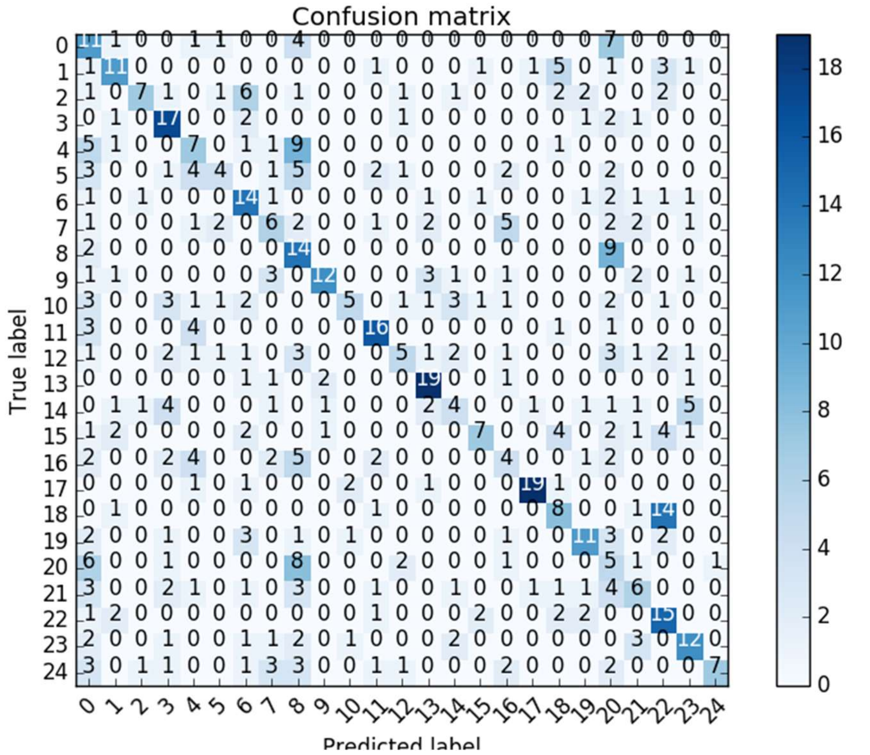
- Examining the logs, maximum accuracy on the test set was:
    - LR=0.01 => **41.92%** after **4400** iterations
    - LR=0.001 => **56.16**%after **10200** iterations
- For learning rate **0.01**, after 4500 iterations, accuracy saturated and did not increase further. It kept floating somewhere between 39% & 41% & after 6000 iterations, it started decreasing on a long term and reached till a low of 16%.
- For learning rate **0.001**, accuracy was steadily non-decreasing till around 10k iterations, and after it reached a final maximum of 56.16%.

<u>Testing</u>

- *Confusion Matrix & Analysis*
    - Accuracy from confusion matrix for both the learning rates are **39.54% & 51.28%.** (It's lower than one in the logs because the snapshot was saved at 4000, 10000 rather than 4400, 10200 iterations.)
    - From both the confusion matrix it looks like classes **Street**(4) & **Sand**(18) performed the **best**; while **Observatory**(24), **ShoeShop**(21) & **DormRoom**(6) performed **worst**.
    - It can be seen that one with LR=0.001, is more concentrated on the diagonal.



CASE A, LR = 0.001, 51.28%



CASE A, LR = 0.01, 39.54%

<u>Case B:</u> <u>With</u> pre-training, **<u>No</u>** extra-images

Only 2500 training images mentioned were used for it & was tested on 625 images.

<u>Training</u>

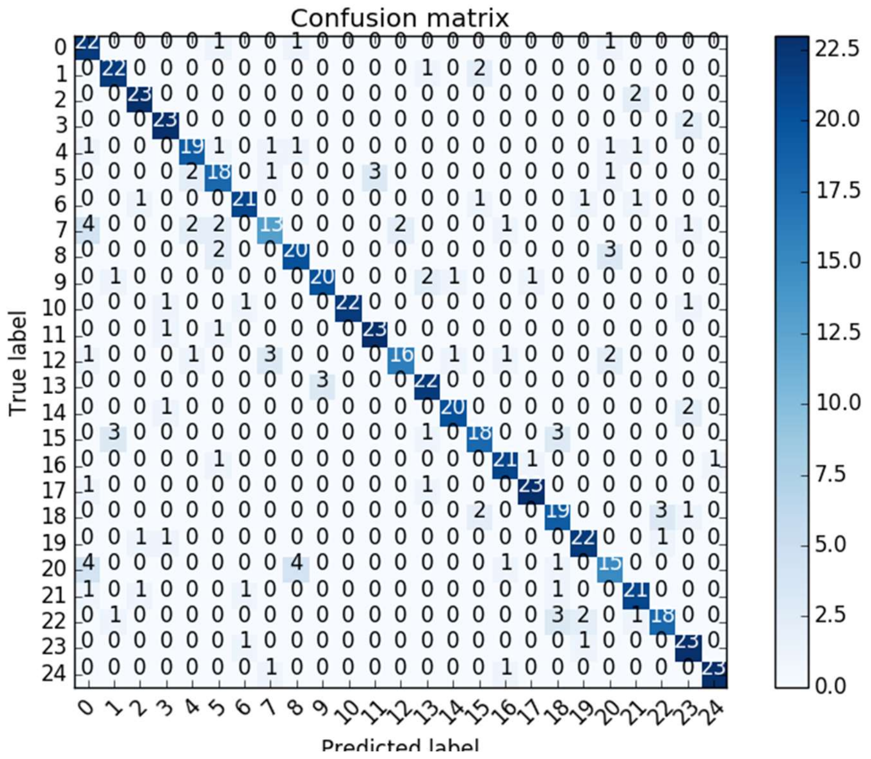
- Network was initialized with **pre-trained weights of ImageNet** using /caffe/models/bvlc_reference_caffenet/ bvlc_reference_caffenet.caffemodel
- **FineTuning** was done with learning rate **0.001**
- For the last layer **fc8**, **LR Multipliers** were kept at **10, 20**; unlike other layers where they were 1,2

<u>Logs</u>

- Examining the logs, maximum accuracy on the test set was:
  - **82.72%** after **2800** iterations
- After the first 200 iterations only, accuracy was at 78%. Over next 5000 iterations, it kept floating between 76% & 83%, with maximum of 82.72% at 2800 iterations

<u>Testing</u>

- *Confusion Matrix & Analysis*
  - Accuracy on confusion matrix is **81.51%**
  - Classes **Hospital**(8), **Operating**(13), **ShoeShop**(21), does not seem to perform well; while almost all are performing very well.



CASE B, 81.51%

Case C: **No** pre-training, **With** extra-images

Other than given 2500 training images, extra 4105 images from the SUN dataset only were used for training, thus a total of 6605 images & was tested on 625 images.

Training

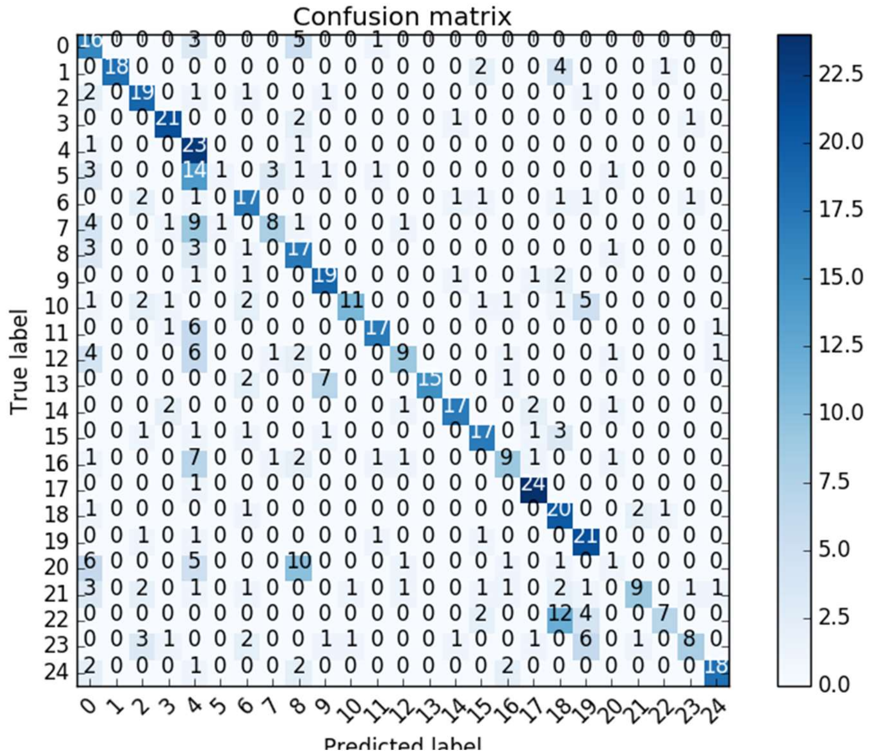
- Network was initialized **randomly**
- Training was done for **two learning rates** – 0.01, 0.001

Logs

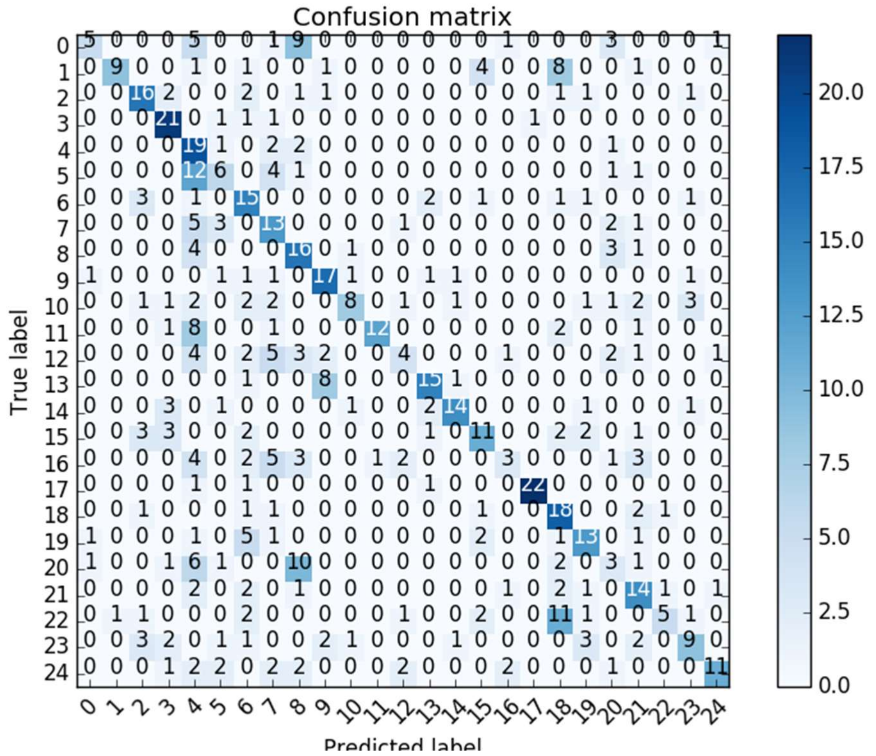
- Examining the logs, maximum accuracy on the test set was:
  - LR=0.001 => **64.48%** after **8400** iterations
  - LR=0.01 => **53.28**%after **5600** iterations
- Overall behavior for both learning rates was similar as in case A.

Testing

- *Confusion Matrix & Analysis*
  - Accuracy on confusion matrix are **58.2% & 48.07%**
  - **Hospital**(8), **Railway**(11) , **Operating**(13), **Garrage**(17), **ShoeShop**(21), **FormalGarden**(23) are performing **worst**, while Dining **Room**(5), **Sand**(18) are performing **best**.
  - Here also, as in class A, it can be seen that confusion matrix with LR=0.001 is more concentrated on diagonal.



CASE C, LR = 0.001, 58.2%



CASE C, LR = 0.01, 48.07%

<u>Case D:</u> <u>With</u> pre-training, **<u>With</u>** extra-images

Other than given 2500 training images, extra 4105 images from the SUN dataset only were used for training, thus a total of 6605 images & was tested on 625 images.

<u>Training</u>

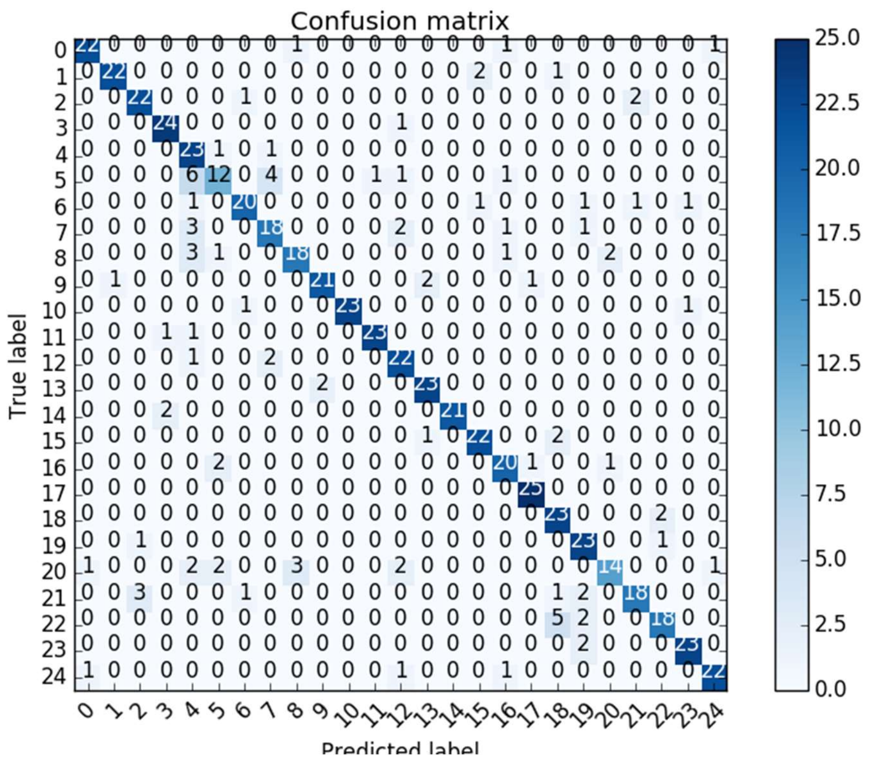
- Network was initialized with **pre-trained weights of ImageNet** using /caffe/models/bvlc_reference_caffenet/ bvlc_reference_caffenet.caffemodel
- **FineTuning** was done with learning rate **0.001**
- For the last layer **fc8**, **LR Multipliers** were kept at **10, 20**; unlike other layers where they were 1,2

<u>Logs</u>

- Examining the logs, maximum accuracy on the test set was:
    - **84.96%** after **10000** iterations
- Again as in case B, because of pretraining, it reached to accuracy of around 78% after 200 iterations only. At 10000 iterations, it reached a maximum of 84.96.

<u>Testing</u>

- *Confusion Matrix & Analysis*
    - Accuracy on confusion matrix is **83.9%**
    - **Here almost every class is performing very well, compared to previous 3 cases**, & everything is concentrated on diagonal.



CASE D, 83.9%

Conclusion:

From these 4 cases, it can be concluded that

- **PreTraining** with ImageNet HELPS A LOT in increasing the accuracy.!
- Also, training with **extra Images** do help up to some level in increasing accuracy.
- While learning from scratch, **learning rate** can have a major impact on final accuracy, as can be seen in cases A & C, with learning rates 0.01 & 0.001.
- Classes such as **ShoeShop, Operating, Observatory** are performing worse consistently, while class **Sand** is performing best.