RCNN for Symmetric Groups

Skanda Bharadwaj

September 14, 2019

The aim of this project is to detect and localize the unit lattices in any given image from the wallpaper group dataset using deep neural networks. In particular, we propose to use R-CNN (Regions with Convolutional Neural Network) to find the unit lattices in any given image.

Contents

1 Introduction

The ability to detect patterns in natural scenes serves critical biological needs. Nevertheless, it poses substantial computational difficulties. Repeating patterns are not uncommon in the nature. In this project we try to identify the very building blocks of these repetitive patterns that are responsible for the symmetry. We will take up the wallpaper dataset to design algorithms to pick up the fundamental units of repetitive patterns. A wallpaper group (or plane symmetry group or plane crystallographic group) is a mathematical classification of a two-dimensional repetitive pattern, based on the symmetries in the pattern. Such patterns occur frequently in architecture and decorative art, especially in textiles and tiles as well as wallpaper. We propose to discover the individual units of these patterns, also called unit lattice, using region proposal networks such as R-CNN or Faster R-CNN.

2 Related Work

There has been significant contributions to different areas of classification using region proposal networks. Also, there has been significant contributions towards understanding the wallpaper group dataset from several standpoints. Both the cases are elaborated in the following sections.

2.1 Related Work on the Wallpaper Groups

Wallpaper groups dataset have been studied to understand the symmetries from different perspectives. Liu et. al. proposed computational model for periodic pattern perception based on the mathematical theory of crystallographic groups. Although, there has been studies on the models of symmetric groups, there's very scarce literature when it comes classification and lattice detection and none of them use region proposal methods to detect unit lattice.

2.2 Related work on the Pattern Recognition Approach

Convolutional Neural networks are being used extensively for classification problem in wide range of applications. R-CNN was first introduced to detect and localize objects in a given scene. Further, networks like faster R-CNN were developed for multi-object detection and localization. The most recent contribution was mask R-CNN, which not only detects and localizes the image, but also creates instance segmentation. In this project, we propose to use R-CNN for the detection of unit lattice.

3 The Wallpaper Group Dataset

Wallpaper group is a two-dimensional repetitive pattern, based on the symmetries in the pattern. There are particularly characterised by subtle differences that may place similar

patterns in different groups, while patterns that are very different in style, colour, scale or orientation may belong to the same group.

The parameters of the wallpaper data set is as follows -

- Number of Classes : 17
- Dimension : 256 x 256 (single channel, greyscale image)
- Number of Training Observations: 340,000
- Observations per class in training dataset:
  - Class 1 through Class 17 : 20,000
- Number of Testing Observations: 51,000
- Observations per class in testing dataset:
  - Class 1 through Class 17 : 3,000

Figure 1 represents the 17 lattices found in the wallpaper groups and a few different wallpaper image instances, where these unit lattices are repeated to create repeating patterns to form symmetric images.

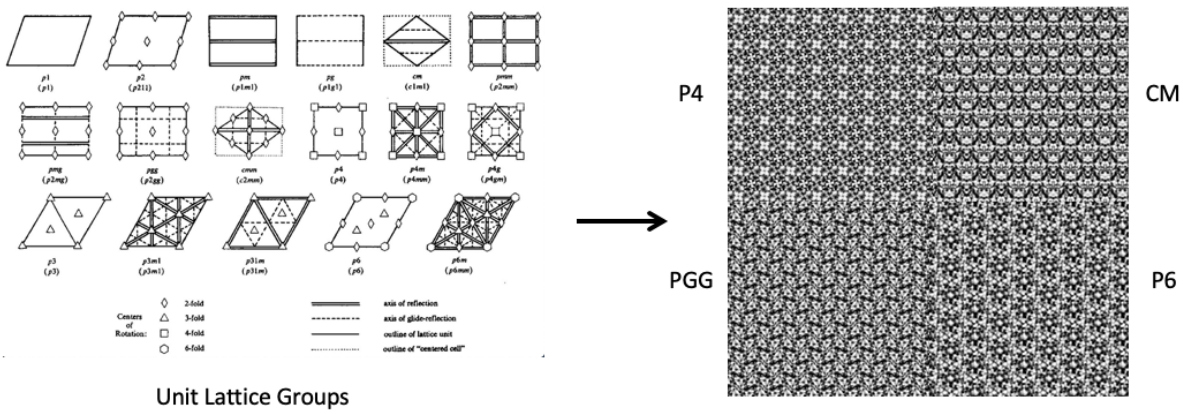


Figure 1: The 17 groups of unit lattices and the images, in which, the unit lattices form repeating patterns.

3.1 How much Training Data is required for this problem?

This is a very common question that rises while solving a problem using machine learning. But the answer majorly depends on the problem statement. But to generalize, it can be said that the following parameters can be taken into consideration before choosing the size of the dataset,

- Number of class

- Number of features
- Inter- and Intra-class variance
- Model for classification

The general rule of thumb is to use 10 times the number of parameters in the learning model. Let us take a look at the aspects of this project. In this project, we proposed to use R-CNN and the we chose pre-trained alexnet to accomplish the goal. As explained in further sections, we decided to make this a 15 class problem as opposed to the ImageNet, a 1000 class problem, the original alexnet was designed to classify. When it comes to alexnet, it has about $6M$ parameters. Since a pre-trained network was used, we only had to change the parameters in the last layer. The total number of parameters in the last layer is $61,455$. According to the rule of thumb, we would be needing $614K$ images. It should also be noted that the images used in this project are greyscale images and not the colour images for which alexnet was originally used. The wallpaper dataset was well studied (from the previous projects) and it was understood that the inter- and intra-calss variance was high enough. Besides these observations, resource constraints were also taken into considerations. With the available time, the memory and the other mentioned aspects we decided to use $304K$ images for training. Although there is no quantitative justification to prove the choice is right, the chosen dataset gave promising results.

4 Convolutional Neural Networks and Wallpaper Groups

This section is devoted to understanding the problem statement more in-depth and the challenges thrown by it. We will discuss the advantages and the disadvantages of the wallpaper groups dataset from the problem solving viewpoint.

4.1 Convolutional Neural Networks as Object Detectors

Convolutional Neural Networks which is a very popular algorithm for image classification and typically comprises of convolution layers, activation function layers, pooling layers to reduce dimensionality without losing a lot of features. Convolutional Neural Nets are mostly used for classification of objects. But in recent years, as mentioned earlier, CNNs have been used to detect and localize objects. In a common scenario, there are different objects in a given image with very distinct foreground to background separation. Figure 2 represents a common scenario in which CNNs are used to classify different objects. It can be observed that, in Figure 2, there is a distinct separation between foreground and background. Other important aspects to be taken in to consideration are the number of objects and the size of the objects. The number of true objects in a given image effects the speed of the algorithm. The lesser the number of object proposals, faster is the classification rate. When the objects are of varying size, selective search algorithms are be used as opposed to the scenario in wallpaper dataset, which will be discussed shortly.

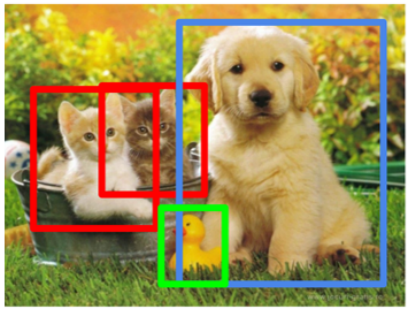


Figure 2: Multi-object detection using CNNs. It can be seen that, in common scenarios, the classes are different and there is a very distinct separation between foreground and background.

4.2 Object Detection in Wallpaper Dataset: An Investigation.

Wallpaper datasets, on the other hand, are, in a way, both easy and hard when it comes to the classification using convolutional neural nets. When it comes to considering the pros of the wallpaper dataset,

- the patterns in any class repeat the same way for all the images in the class. Therefore, the unit lattices can be easily manually labelled and considered for training the CNN.
- the class are well separable and the classification accuracy can be high.

While the above-mentioned points are in favour of the wallpaper dataset, the following are the major drawbacks of the wallpaper dataset.

- Given any image from the wallpaper dataset, the entire image is filled with objects and more importantly, object belonging to the same class.
- The patterns can start from anywhere in the image and continue to the repeat throughout the image. Hence, selective search cannot be performed to create the region proposals for classification.
- There is no background for any given image. Therefore, every search becomes a proposal for classification that increases the computations by a large margin.

After investigating the advantages and disadvantages of the wallpaper dataset, the R-CNN pipeline was custom designed to detect and localize the unit lattices in the wallpaper dataset.

5 Experiments

5.1 Transfer Learning using Pre-trained Alexnet

In this project, pre-trained alexnet was used to learn the unit lattices. Each image in the wallpaper was manually labelled to identify the unit lattices. Each of these fundamental units are extracted and are resized to input size of the alexnet, which is $(227, 227, 3)$. A database of these unit lattices of size $(227, 227, 3)$ is created. This is represented in Figure 3.

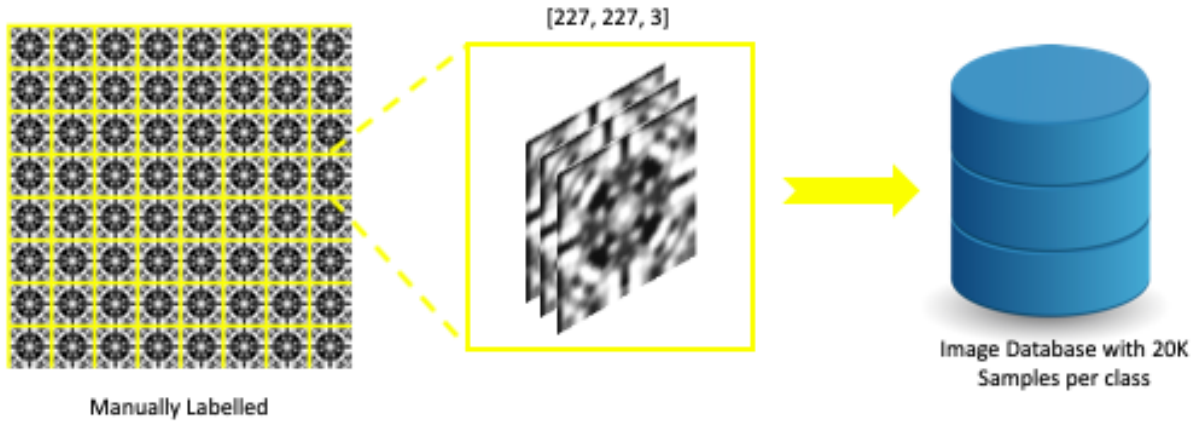


Figure 3: Given any image, unit lattices are manually labelled and are converted into $(227, 227, 3)$. These $(227, 227, 3)$ sub-images from all the images are used as the database to train alexnet.

Figure 4 represents the schematic of the classification pipeline. The created database in the previous step is used to train the pre-trained alexnet. The input of the alexnet is $(227, 227, 3)$. Feature extraction is done through five convolutional layers. A total of 4,096 features of each unit lattice is collected from the last convolutional layer to input to the fully connected layer. Three fully connected layers are used to learn the classes of unit lattices. The first two layers consists of 4,096 neurons and the number of neurons in the last fully connected layer is equal to the number of classes. A total of 18,000 images for training, 2,000 images for validation and 3,000 images for testing were used per class. Alexnet was trained for 15 classes (eliminating P6 and P6M, since manual labelling took too much time). Therefore, a total of 270,00 images were used for training, 30,000 images for validation and 45,000 images for testing of the pre-trained alexnet. Each image represents unit lattice and the alexnet is trained to classify these unit lattices with a modification to the last fully connected layer to match to the number of classes of unit lattices.

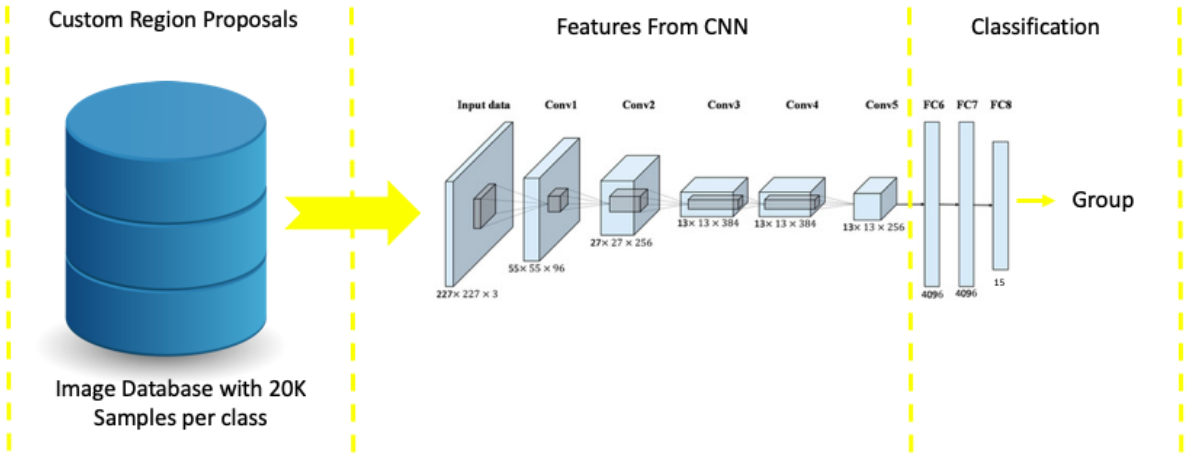


Figure 4: Given any image, unit lattices are manually labelled and are converted into $(227, 227, 3)$. These $(227, 227, 3)$ sub-images from all the images are used as the database to train alexnet.

5.2 Detection and Localization of Unit Lattices in Test Images

The previous section only addressed the classification problem. The final goal to detect, classify and localize all the repeating patterns in a given image. As mentioned earlier, we face three major problems while dealing with the the wallpaper images. Given below are the problems that arises with each of the problems.

- An image consists of only object. Therefore, selective search is not possible and the only option left is the brute force search.
- All objects belong to the same class and, therefore, every search has a very high confidence value. Every almost search produces a true-positive classification output creating bounding boxes every search. Non maximum suppression is required at every stage in order to eliminate the overlapping bounding boxes.
- Since there is no background-foreground separation, bounding box regression cannot be performed.

In order to detect and localize the unit lattices in the wallpaper images, a pipeline was designed as shown in Figure 5. Every test image is of the size $(256, 256)$. A sliding window of predetermined size (w, h) is iterated over every row and column. This is referred to as brute force search to find the unit lattice. It can be easily seen that the number of region proposals created per image are $(256 - w)\text{x}(256 - h) \approx 65k$. Each of these search regions is sent into the classifier net for classification. Since all the search regions belong to the same class, a very high threshold is set to select the given the region as object. It was found that, despite setting a very high threshold, a large number of search regions were selected as true-positives, which is an expected behaviour given the nature of the wallpaper dataset. In order to eliminate overlapping bounding boxes,

non-maximum suppression was implemented. Figure 6 represents an example of non-maximum suppression. On the left side, it can be seen that a face is detected with three selected true-positive search regions. On the right hand side, the three regions are reduced to a single region true-positive region.

Figure 5: Detection and Localization Pipeline

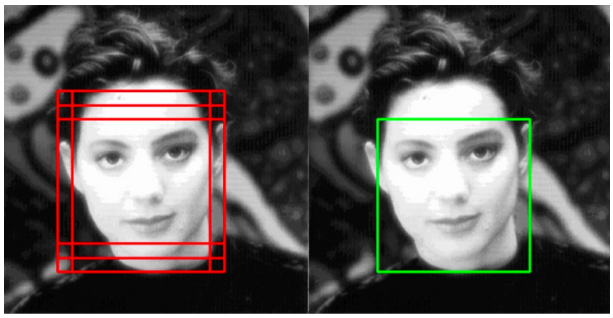


Figure 6: Non-Maximum Suppression

6 Results and Discussion

6.1 Unit Lattice Classification Using Pre-trained Alexnet

As mentioned earlier, alexnet was used for the classification of the unit lattices. Figure 7 represents the architecture of the pre-trained alexnet that was used for classification. Table 1 represents the configuration parameters and overall results of the alexnet.

Parameters	**Alexnet**
Learning Rate	1E-04
No. of Epochs	10
Batch size	400
Training Accuracy(%)	99.83
Validation Accuracy(%)	99.77
Testing Accuracy(%)	96.39
Training Time (hr)	11.5

Table 1: Configuration Parameters and Overall Results of Alexnet.

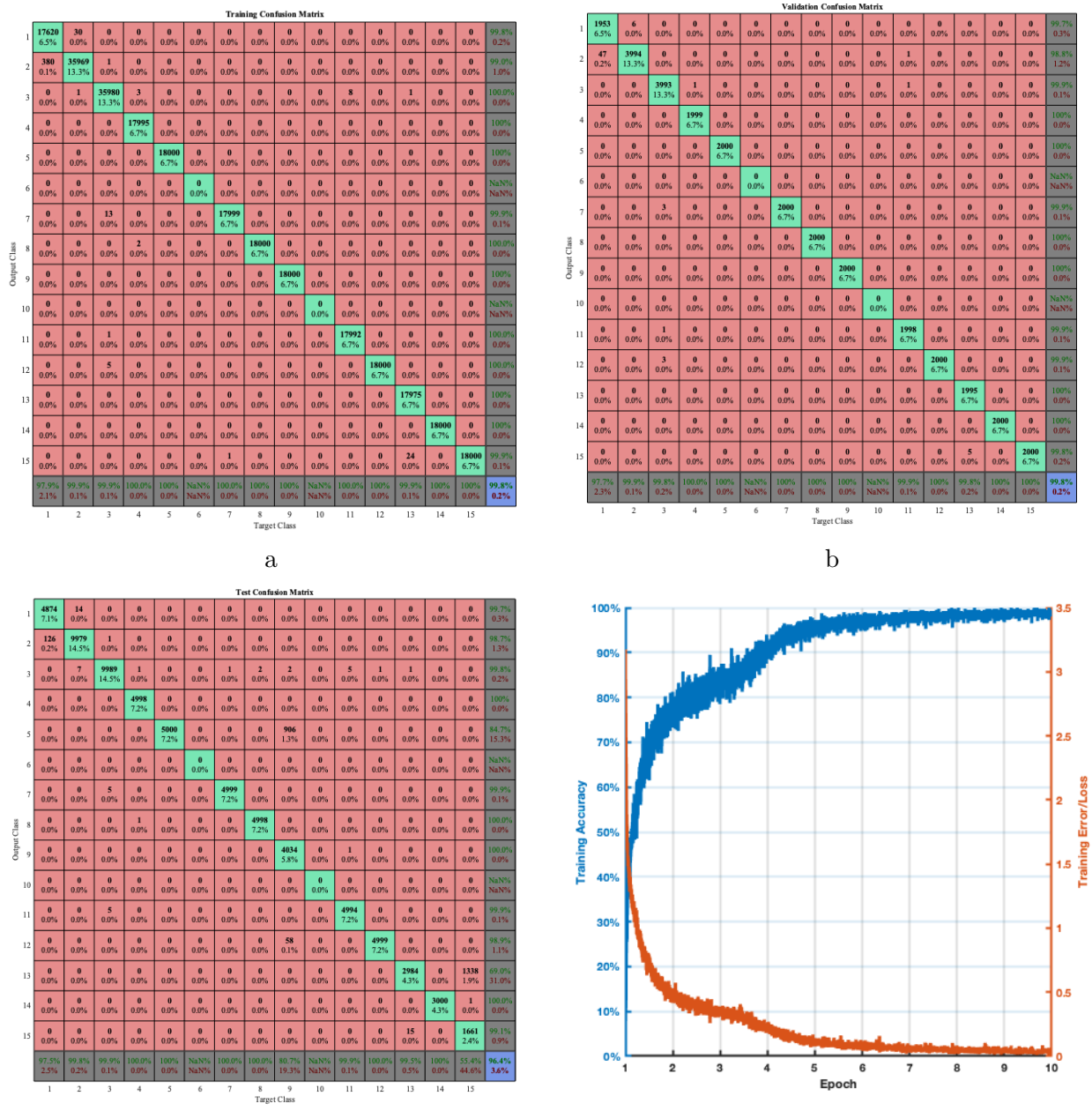
Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	227x227x3	-	-	-
1	Convolution	96	55 x 55 x 96	11x11	4	relu
	Max Pooling	96	27 x 27 x 96	3x3	2	relu
2	Convolution	256	27 x 27 x 256	5x5	1	relu
	Max Pooling	256	13 x 13 x 256	3x3	2	relu
3	Convolution	384	13 x 13 x 384	3x3	1	relu
4	Convolution	384	13 x 13 x 384	3x3	1	relu
5	Convolution	256	13 x 13 x 256	3x3	1	relu
	Max Pooling	256	6 x 6 x 256	3x3	2	relu
6	FC	-	4096	-	-	relu
7	FC	-	4096	-	-	relu
Output	FC	-	15	-	-	Softmax

Figure 7: AlexNet Architectural Details

In Figure 8 a, b, c represent the confusion matrices of training, validation and testing respectively. d represents accuracy and loss over epochs. In the confusion matrices, the rows correspond to the true class and the columns correspond to the predicted class. Diagonal and off-diagonal cells correspond to correctly and incorrectly classified observations, respectively. The column on the far right of the plot shows the percentages of all the examples predicted to belong to each unit lattice group that are correctly and incorrectly classified. These metrics are often called the precision (or positive predictive value) and false discovery rate, respectively. The row at the bottom of the plot shows the percentages of all the examples belonging to each unit lattice group that are correctly and incorrectly classified. These metrics are often called the recall (or true positive rate) and false negative rate, respectively. The cell in the bottom right of the plot shows the overall accuracy. These are per-group result summary.

6.2 Detection and Localization of Unit Lattices in Test Images

For the detection and localization of the unit lattices in a given image, a brute force search is first carried out. A sliding window is moved through all rows and columns. All the pixels inside the sliding window constitutes a sub-image. Each of these search regions that forms a sub-image is, then, converted into a $(227, 227, 3)$ image and classified using alexnet to predict the group represented in the sub-image. A very high threshold on the confidence of the prediction is set to select a given sub-image as representing the unit lattice. The threshold is set very high since all the sub-images belong to the same class. Once a given sub-image is selected as representing a unit lattice, a bounding box is drawn over the sub-image. Given that the search is brute force, it can be easily seen that the more than one sub-image will be selected as representing the unit lattice for the same unit lattice in the original image. Therefore, non-maximum suppression is implemented



a b c d

Figure 8: Figures a, b and c represent confusion matrices of training, validation and testing respectively. d represents accuracy and loss over epochs, the learning rate.

to mitigate the issue of bounding box overlap. Figure 9 represents the outputs of the detection and localization pipeline. In the left images, the green bounding boxes represent the manually labelled unit lattices. In the centre images, the red bounding boxes represent all the selected sub-images as representing unit lattices. In the right Images, the yellow bounding boxes represent the unit lattices after non-maximum suppression.

Similarly, Figure 10 represents the detection and localization of unit lattices in several

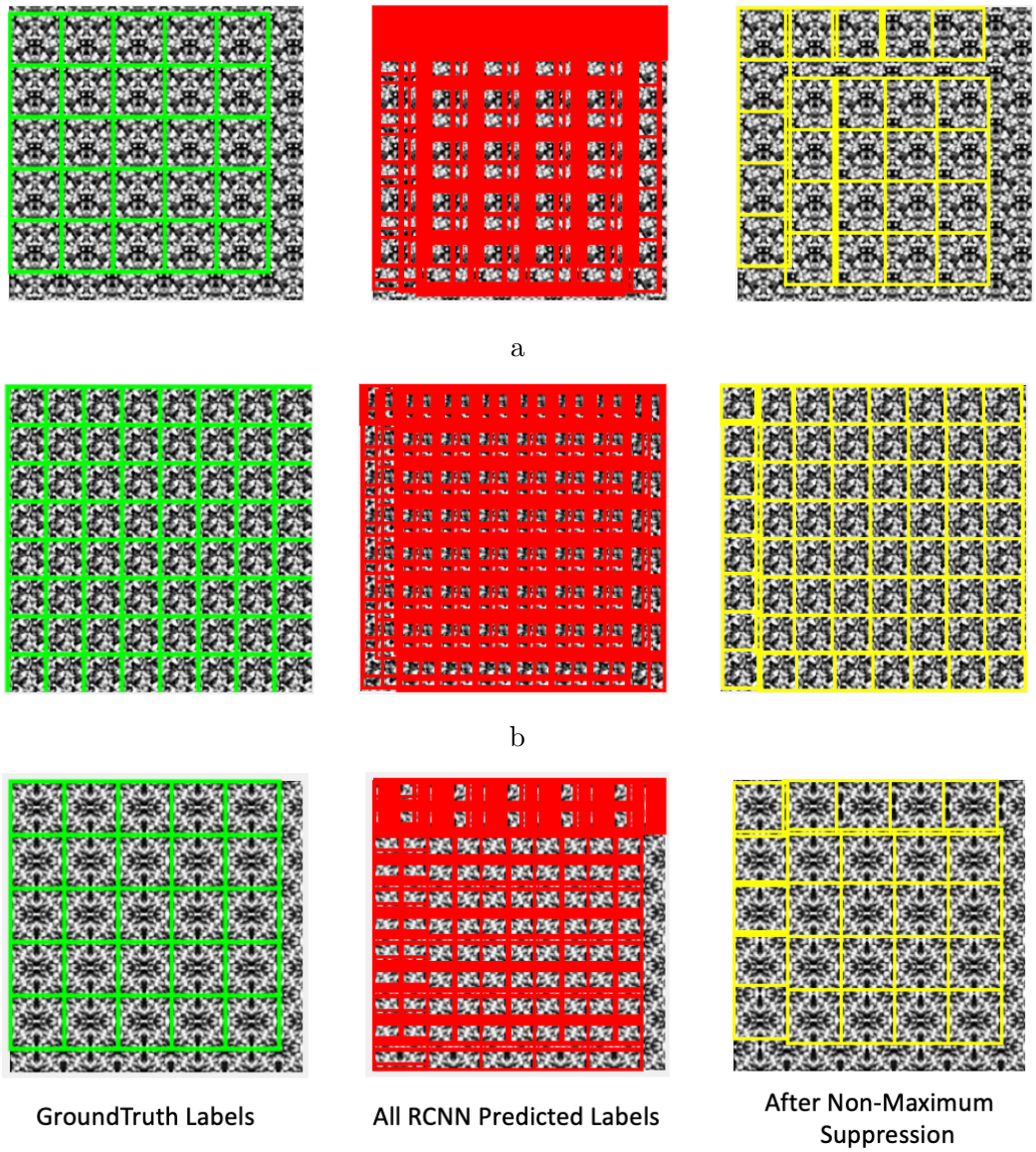


Figure 9: Images to the left represent the manually labelled unit lattices in a given image. Red bounding boxes in the centre images represent all the selected sub-images. Yellow bounding boxes in the right images represent the unit lattices after non-maximum suppression.

images belonging to different groups. It can be observed that detections and localizations for the groups $PMG, P4G, PMM$ and $P4$ are exceptionally good. But the algorithm suffers a bit to precisely identify unit lattices for the groups $P2, CM, P1$ and CMM. Other classes have not been included intentionally since they produce very similar results. It is also worth noting that the network has learnt symmetry in that the algorithm captures the symmetry even though they are not precisely the unit lattices of the group.

For instance, consider the group CMM from the Figure 10. The very first bounding box, i.e. the top left corner, represents the unit lattice. But the ones in the middle have been translated, yet capturing symmetric patterns.

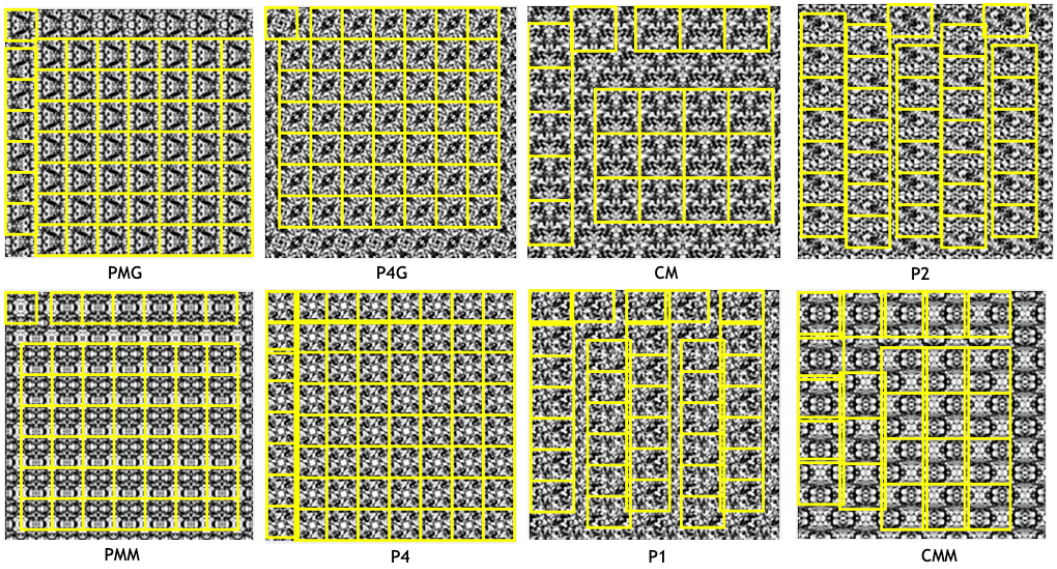
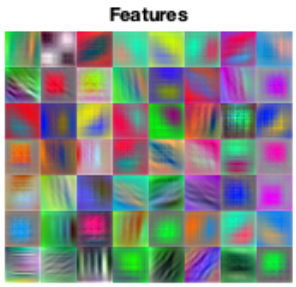


Figure 10: Detection and localization of unit lattices in different groups.

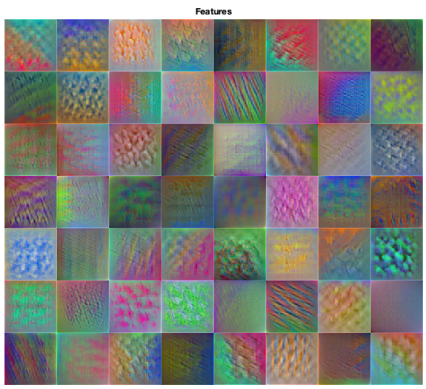
6.3 Visualization of AlexNet Filters and t-SNE

6.3.1 Filters Visualization

Figures 11 and 12 represent the visualizations of the filters of all the convolutional layers of the alexnet, respectively.



a

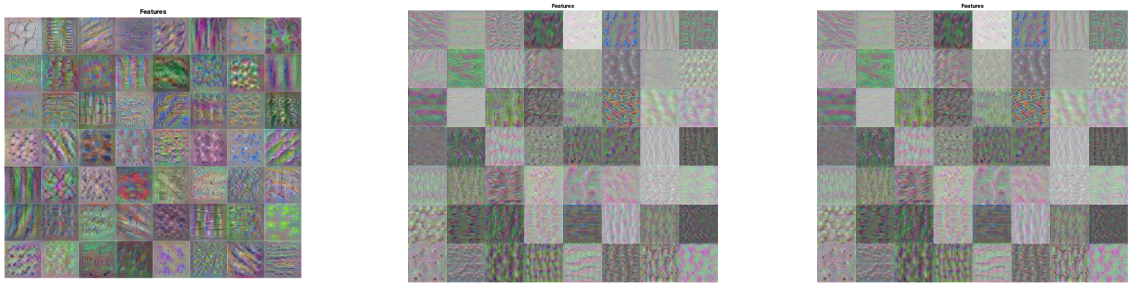


b

Figure 11: Filters Visualization of Convolutional Layers 1 and 2 in AlexNet

It can be seen that the filters in the initial convolutional layers learn fundamental features from the images such as edges, and the filters in deeper convolutional layers

learn more complicated features such as recurring patterns in the images.



a b c

Figure 12: Filters Visualization of Convolutional Layers 3, 4 and 5 in AlexNet

6.3.2 t-Distributed Stochastic Neighbour Embedding

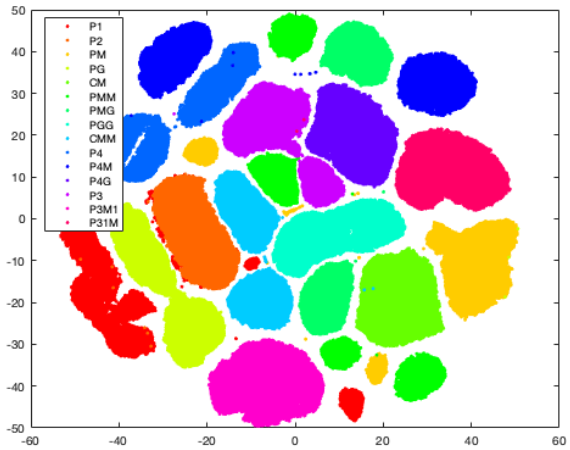


Figure 13: t-SNE from the last fully connected layer of the alexnet.

t-Distributed Stochastic Neighbour Embedding (t-SNE) is a technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets. t-SNE was visualized for the alnexnet used for training the unit lattices. Figure 13 represents the t-SNE from the last fully connected layer of alexnet (23^{rd} layer). From the Figure 13 it can be observed that the inter-class variance is high and, as a result, the separation between the groups is quite distinct. This justifies the high classification accuracy of the system.

7 Conclusions

Although the classification results were extremely good and the detection and localization of the unit lattices was considereably reasonable compared to the ground truth labels,

given below are a few findings from the experiments.

- Number of region proposals per image = $\sim$ 256x256 $\approx 65k$
- Time for detection of all unit lattices in 1 image is $\sim$ 5mins
- Time for testing on 150 images (10 images/class) $\sim$ 13hours

It should also be noted that these statistics were obtained by running the algorithms on a 4GB GPU. Looking at the above statistics, it can be concluded that R-CNN might not be best suited for the identification of unit lattices in the wallpaper dataset due to the inherent nature of the images as pointed out in the previous sections. May be, classical computer vision techniques might work faster than the deep neural networks.