# CS 335 : Lab-5

Akash Reddy G - 190050038

November 20, 2021

## Contents

# 1 Directed Graph Models

## 1.1 D-Seperation

Two nodes are said to be dependent if there exists atleast one path connecting the two nodes, such that on this path, every tail-tail, head-tail node is not known and every head-head node is known.

Following this simple rule to decide on independence we conclude:

| Statement | Independet | Reason |
|---|---|---|
| $C \perp B$ | No | path CAB is unblocked |
| $C \perp B \vert A$ | Yes | CAB is blocked by A. All other paths are blocked since there's a head-head node on each of them |
| $C \perp B \vert A, J$ | No | path CDFJGB is unblocked since J is a head-head node |
| $C \perp B \vert A, J, D$ | Yes | CAB is blocked by A, CDxxx is blocked by D, CEHxxx is blocked by H |
| $C \perp G$ | No | CDG is unblocked path |
| $C \perp G \vert B$ | No | CDG is unblocked path |
| $C \perp G \vert B, D$ | Yes | CABG is blocked by B. CDxxx is blocked by D. CEHxxx is blocked by H |
| $C \perp G \vert B, D, H$ | No | CEHFG is unblocked as H is head-head node |
| $C \perp G \vert B, D, H, E$ | Yes | CABG in blocked by B. CDxxx is blocked by D. CExxx is blocked by E |
| $B \perp I \vert J$ | No | BGJFI is unblocked as J is head-head node |

## 1.2 Probability Distribution

Joint probability expression can be obtained by multiplying conditional probability of each node given its parents nodes. Hence,

$$P(A,B,C,D,E,F,G,H,I,J) = p(A) \cdot p(B|A) \cdot p(C|A) \cdot p(D|C) \cdot p(E|C) \cdot p(F|D) \cdot p(G|B,D,F) \cdot p(H|E,F) \cdot p(I|F) \cdot p(J|F,G)$$

## 1.3 Number of parameters required to learn the probability distribution

Since the node-I can be found given its parent node-F; and node-H can be found given its parents E,F we already have the conditional probabilities for these 2 nodes.

The number of parameters for a conditional probability of a node is exponential in number of its parents. Here, since bernoulie distribution has only single parameter, we say number of parameters for a conditional probability is equal to 2 raised to number of parents of the node.

| **Prob. distribution** | **#Parameters** |
|---|---|
| $P(A)$ | 1 |
| $P(B\|A)$ | 2 |
| $P(C\|A)$ | 2 |
| $P(D\|C)$ | 2 |
| $P(E\|C)$ | 2 |
| $P(F\|D)$ | 2 |
| $P(G\|B, D, F)$ | 8 |
| $P(H\|E, F)$ | 0 |
| $P(I\|F)$ | 0 |
| $P(J\|F, G)$ | 4 |
| $P(A, B, C, D, E, F, G, H, I, J)$ | 23 |

# 2 CNN Theory Questions

## 2.1 Task 1

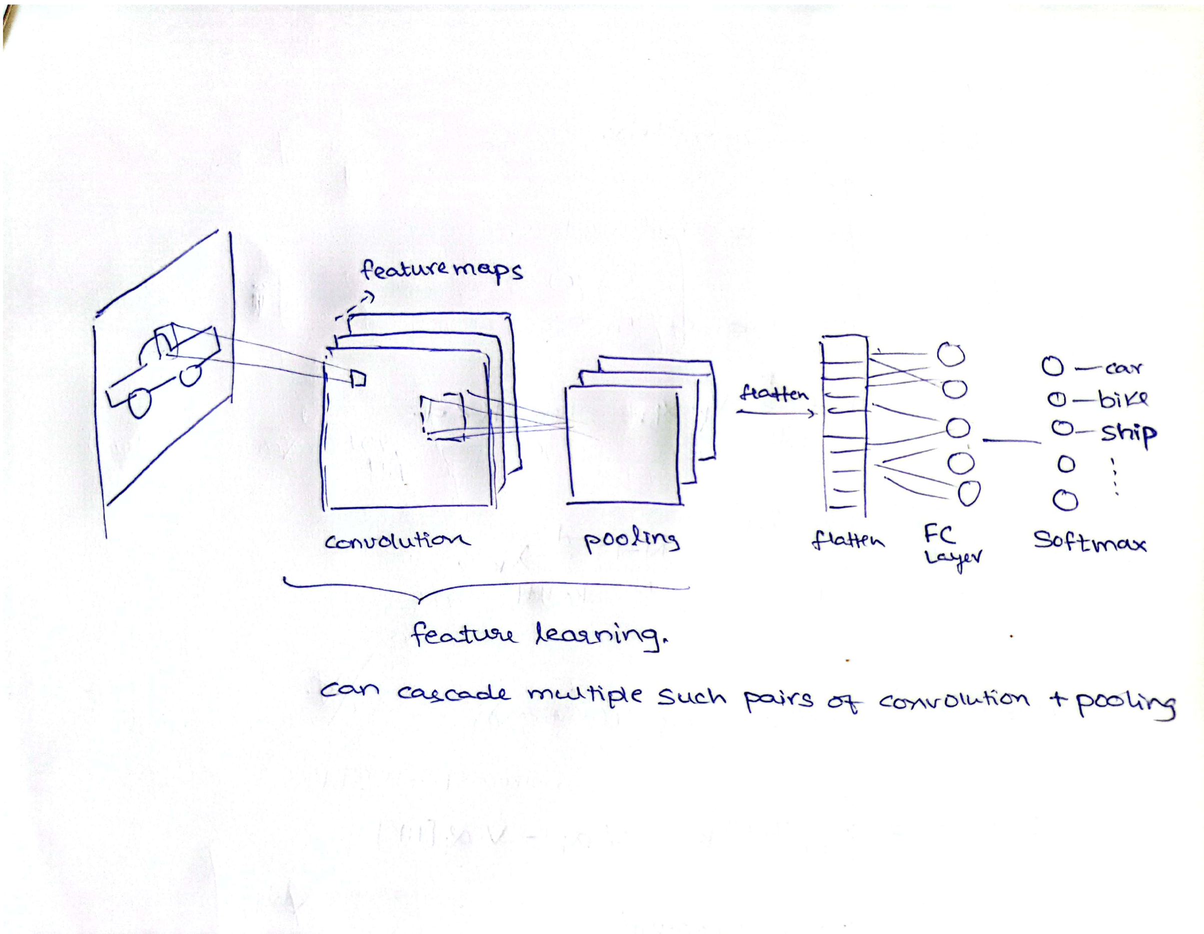
CNNs are particularly useful in image classification as they detect features in an image. The general architecture of CNNs include a convolution layer, a pooling layer (pair of these can be cascaded one after another) and a classification part of the network using feed forward architecture and softmax classification.

One important detail is how we use multiple channels for convolution. Each channel corresponds to a different patch and can be interpreted to capture differnt aspect in the image.
For example, one channel can capture the tyres in the image. Another can capture the windshield in a car. Another would capture the bare engine of a motorcycle. Another for the exhaust of the motorcycle. etc

These features get detected by convolution layer and stay filtered by pooling till the end of the network and activate the relevant vehicle class in the softmax classification. For example, (tyres+engine+exhaust) will activate the motorcycle class; (tyres+windshield+bumper) will activate the car class.

The convolution layer will extract such features from overall image. The pooling layer will filter these higher activations and discard the lower acitvations. This is why, CNNs are able to handle differences in object's location and size.

For example, consider the images of car and bike given in Fig.3 of lab statement. Say a channel detects the windshield of the car and another channel detects tyres of a car. Regardless of whether the car is in the top-left or bottom-right, it is captured by the convolution layer (as we move the patch over the whole image), and these activations are filtered by the pooling layer. After 2-3 times of such convolution and pooling, we'd have the features of a car detected, regardless of its posiiton. These will inturn activate the CAR class in softmax classification.



### 2.2 Task 2

Let us say we have a trained CNN with us, which can classify a 1024x1024 image to say whether it is a car/bike/plane etc. Our current problem is to identify different objects in same image which are well seperated without any occulsion.

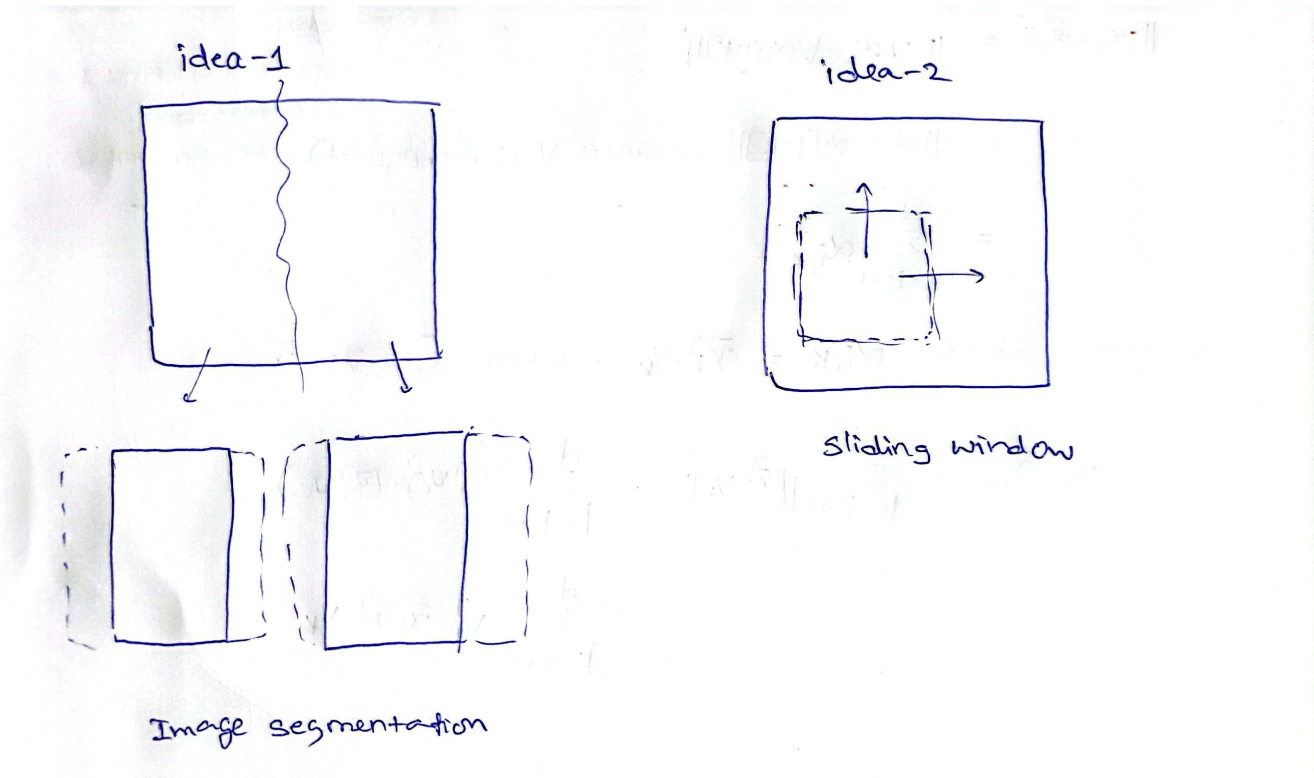
One simple idea is to break the bigger image into segments (say 2 images of 512x1024/ 1024x512 or 4 images of 512x512) and pad them to size 1024x1024 and apply our CNN on individual segemnt. The given two images in Task-2 statement can be distinguished by breaking the given image into 2 images of 1024x512 and padding them and using CNN on them.

This idea will not be robust and might not always identify all objects completely.

Another scheme is to use a sliding window over the image and classify each window using the CNN and give it a "score" based on the softmax coefficent outputted. Later, we will perform a non-maximal supression to determine the number of objects in the image.

Say we use a sliding window of 512x512 with stride of 10 pixels. After obtaining a window, we pad it to 1024x1024 size and classify it using CNN. Then we threshold on the softmax confidence obtained to dicard meaningless classifications. We should be having clusters, and the number of such clusters will be the number of individual objects in the image (since we assumed well-seperated objects in the image) . Then we perform non-maximal supression to obtain that one window, which has highest softmax confidence among its neighbours.

In this manner, we can detect multiple objects in an image using a sliding window algorithm. One important limitation here is the high computation cost of running our CNN on many windows.



### 2.3 Task 3

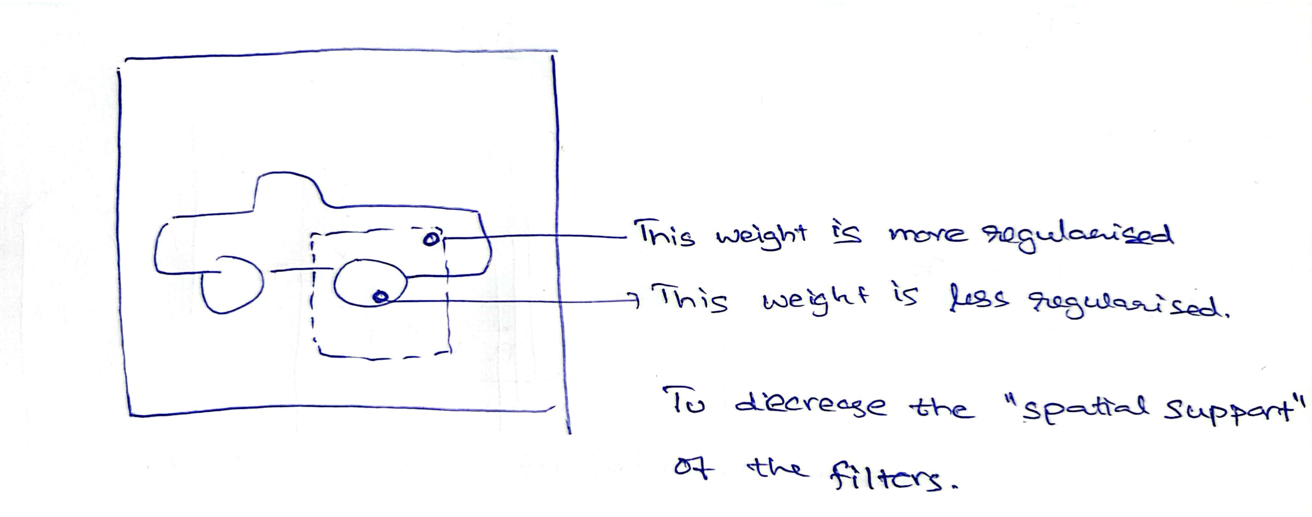
One solution to classify occluded vehicles is to train our original CNN with occluded samples. But this might not always be feasible and hence, we shall take it that we have data of non-occluded objects.

Since occluded images donot fully have the spatial features of clear images, it makes sense to **decrease the range of "spatial detection"** of our patches. This can be obtained by regularizing the weights corresponding to farther spatial location. Hence, we use penalty terms for weights corresponding to father pixels in our loss function. In this manner, we can decrease the "view" of a convolution filter.

In this manner, we obtain channels that correspond to only the front half of a car, only the top of a autorikshaw, only the headlight of a bike, etc

Now, we will apply the same sliding window approach as before to detect objects in the image, using the above trained CNN model. For the first image, we can use bigger patches, but for the 2nd image, we have to use small size patch due to the density of the objects present in the image.

After calculating the softmax scores, we will perform thresholding and then maximal supression to get the final count of objects and their labels.




An important limitation is how this training of the CNN using reduced spatial support filters could result in wrong classification very often. Since occlused objects have very few distinguishing features, its hard to differentiate between them. We will need more deeper CNNs (and more parameters asks for more data to train on) to get better results, when compared to the usual images classification.