Convolutional Neural Network

Computer vision

Edge detection

More edge detection.

Use of edge detection

We use many 0-32 filter to detect picture

Edge detection

edge

Picture matrix * matric ⟹ new matrix

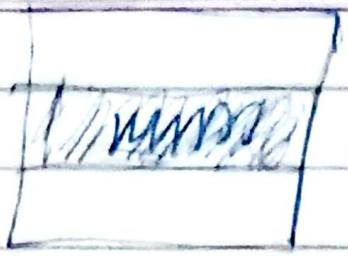
Through this matric machine learn where the pixels are located and learned the patterns.

1	0	-1
1	0	-1
1	0	-1
1	0	-1

Left dark bottom light right dark

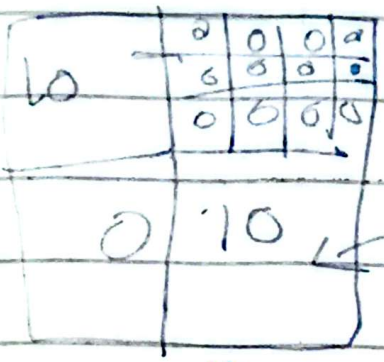
1	1	1
0	0	0
-1	-1	-1

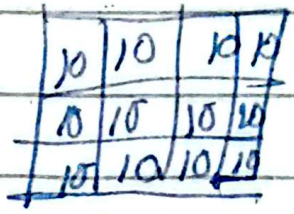
Horizontal



horizontal clavde

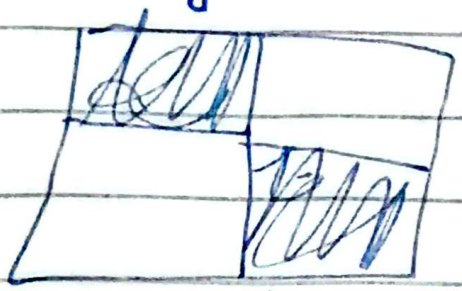
line detected





y

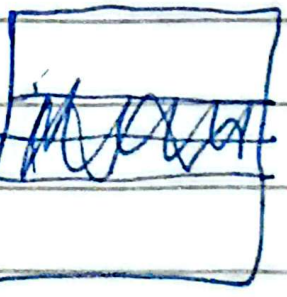
X



edge we ~~get~~ get in pixels

$y \times X =$

0	0	0	0
30	10	-10	-30
30	10	-10	-30
0	0	0	0



Other filters

1	0	-1
2	0	-2
1	0	-1

3	0	-3
10	0	-10
3	0	-3

Sobel filter — Scharr filter

vertical dark

Sobel	Scharr
Basic edge detector feature extraction	Same as sobel. Better at edge detection, regardless of direction Scharr is more accurate.

Padding

When we mark or filter the size of picture shrinks and sometimes u don't want to lose them

* Cause only center things are detected not on the border.

Padding

we get the same size of image where borders are detected.

6×6 * 3×3 *Padding = Pad 6×6

How much pad:

Valid convolution : no padding ~~filter~~ : $6 \times 6 * 3 \times 3 \Rightarrow 4 \times 4$

Same convolution : output size will be same in input and output size.

$\Rightarrow$ $P = 2$ will be required to get full picture.

Strided convolution

*Reduce output size :-

feature maps shrinks, but with learning weights.

Reduce computation

Compute larger patterns fast.

Working

stride = 2

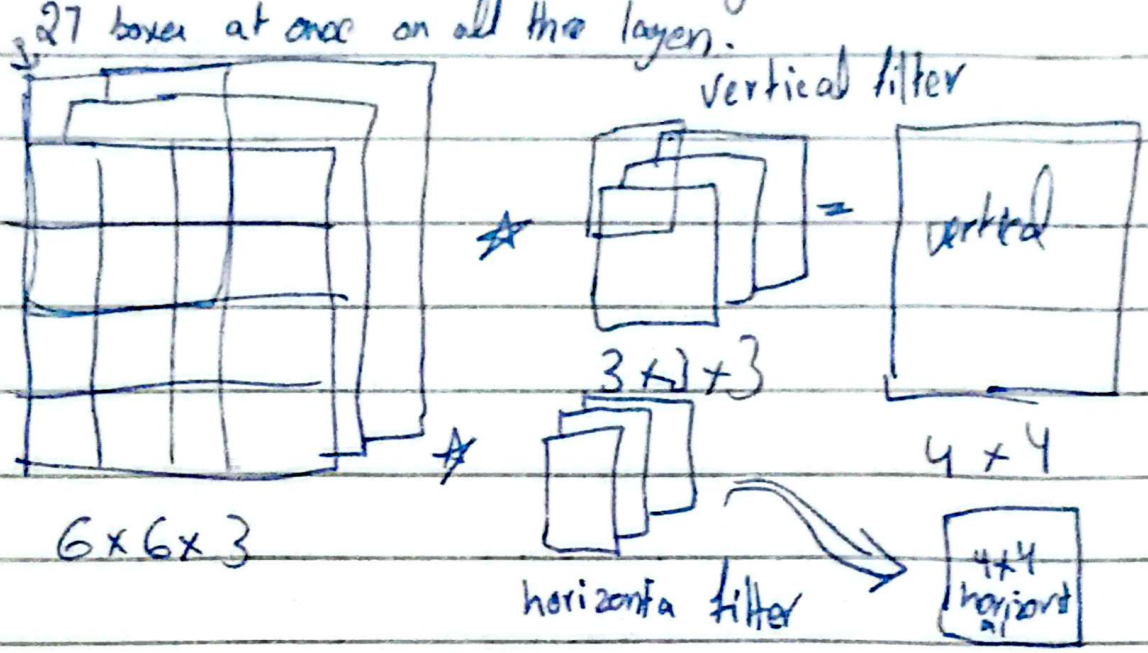
filters ~~jump~~ jump 2 columns in every turn.

By stride

$7 \times 7 * 3 \times 3 = 3 \times 3$

$$\left[\frac{n + 2p - f}{s} + 1\right] \times \left[\frac{n + 2p - f}{s} + 1\right]$$

Convolution over RGB image



work as normal filter but compute all three RGB layer at once and add them the move to next rows or colump

One layer of a convolutional network

If layer l is a convolutional layer,

$f^{[l]}$ = filter size

$p^{[l]}$ = padding

$s^{[l]}$ = stride

If $n^{[l]}$ = number of filters.

Each filter is:

Activations:

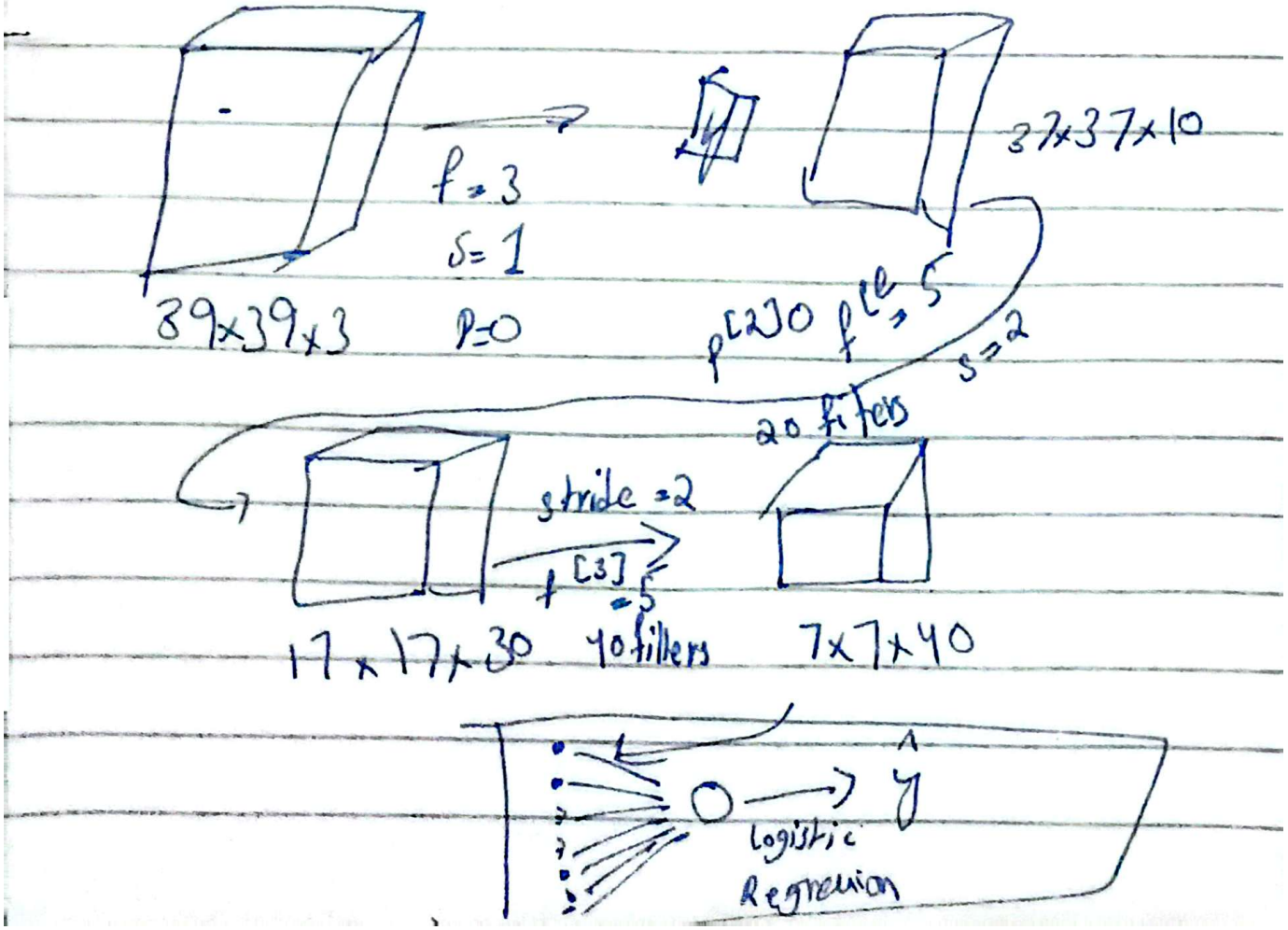
Weights:

bias:

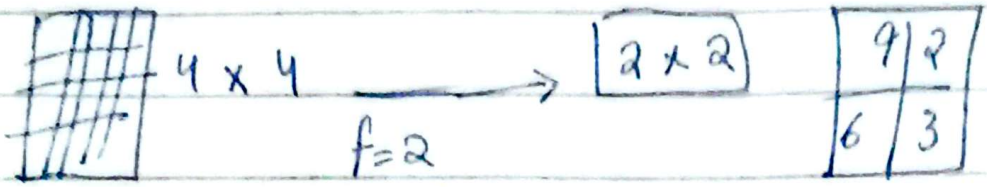
Simple Convolutional Network example:-

p=0 means valid convolutions. . . . . .

Conv Net



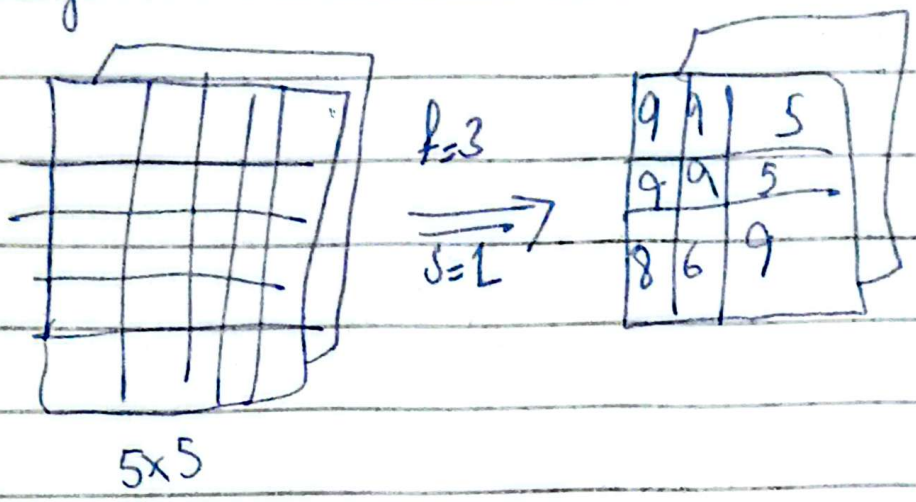
Pooling Layer : Max pooling



Hyper parameters:

It extract features and reduce size of image.

e.g



This takes maximum number in a filter.

Type: Average filter

this take average in a filter

Result:

Max pooling is used more than the average pooling.

Summary

- Convolution layer : learns fillers or to detect edges filter, corners.
- Pooling layer:-

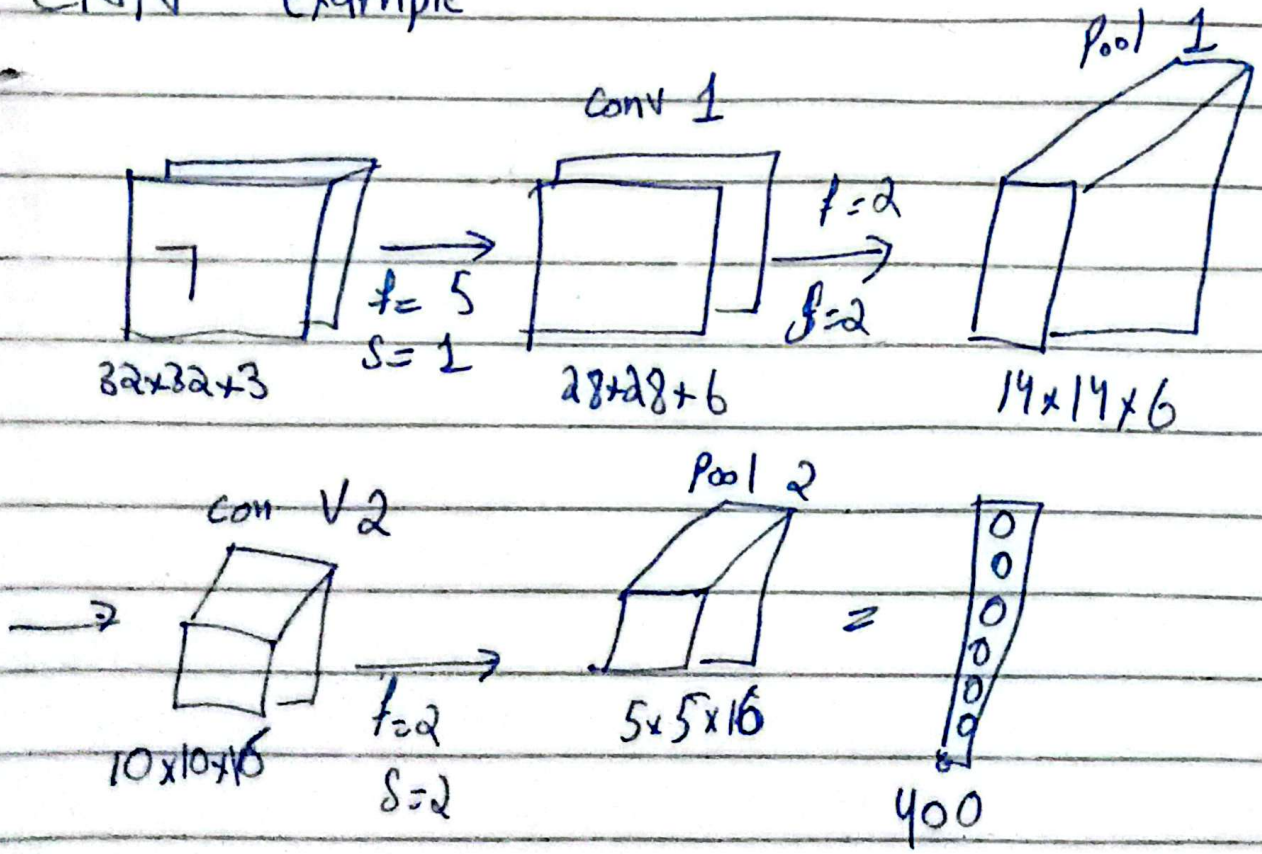
Makes images smaller while keep important info- Reduces computation

Fully connected layer:-

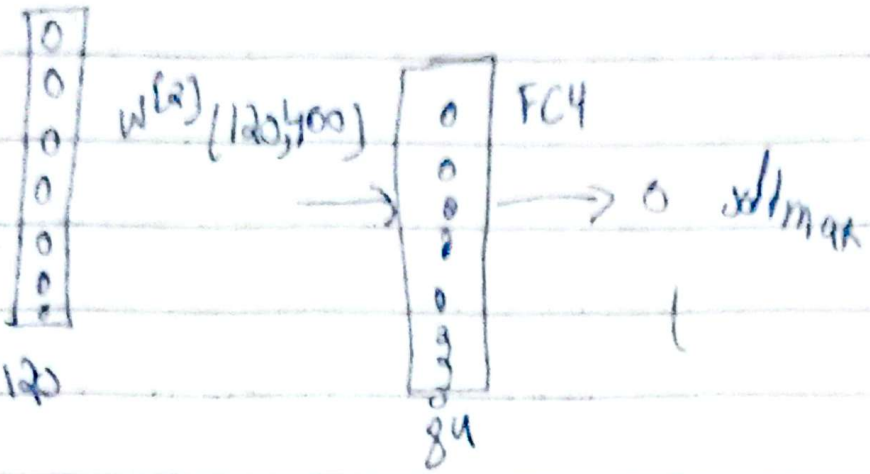
work: on normal NN

use activation function features to predict.

CNN example

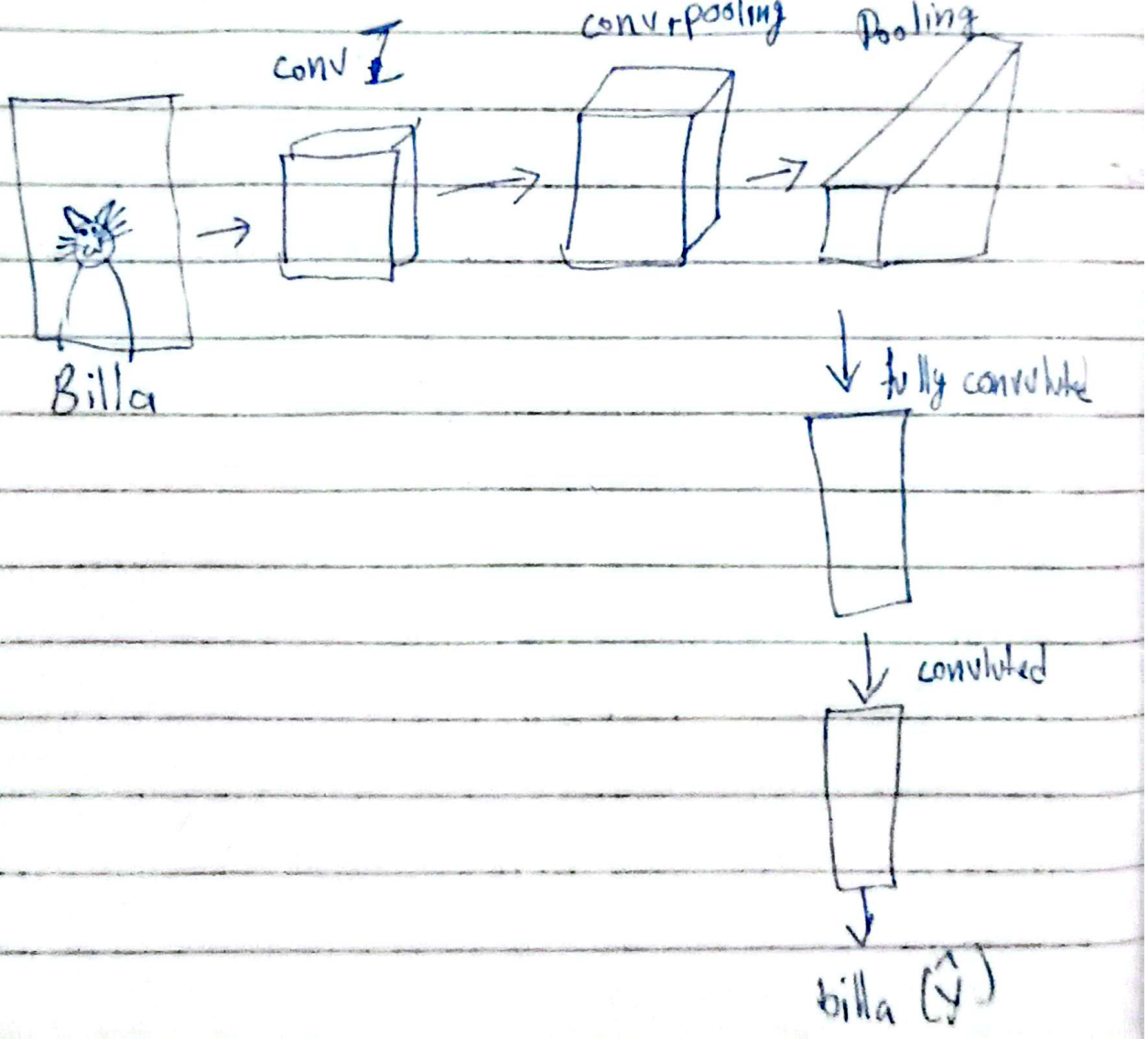


FC3

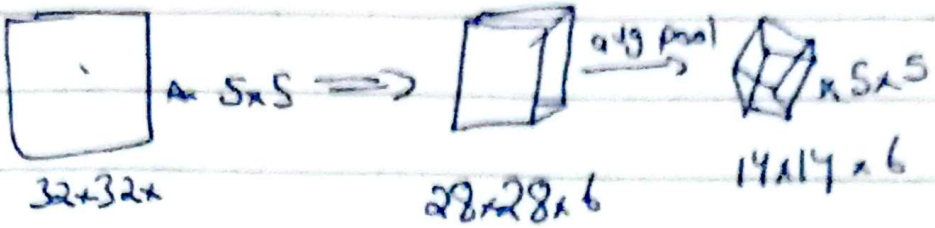


Research:-

1# Why convolutions:-



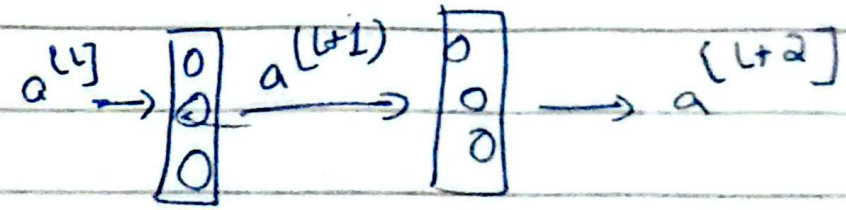
Old Net (Le Net) 1998



VGG16 (16 layer)

Alex Net

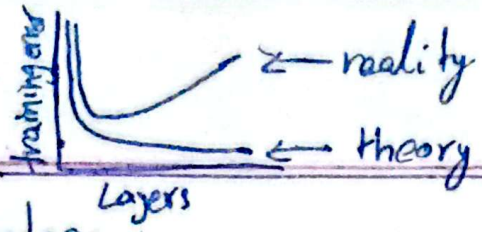
Real Resnet



$a^{[l]}$ → linear → Relu → $a^{[l+1]}$ → linear → Relu → $a^{[l+2]}$

⟹ Resnet allow u to train much deeper

Graph

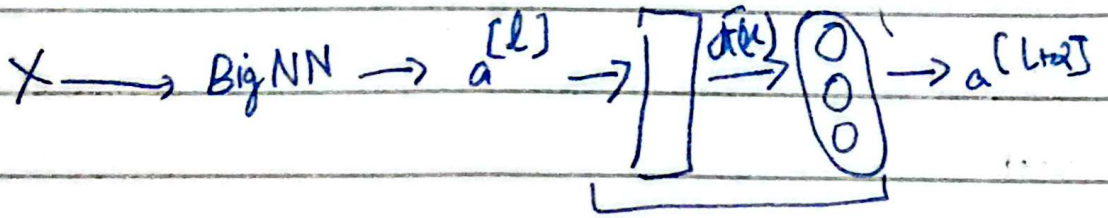


Disadvantage

1) High Computational Cost
1) Overfitting on small dataset
3) ~~Only~~ Extremely deep versions give only marginal accuracy gain while increasing training cost.

4) Deeper model are slower during prediction

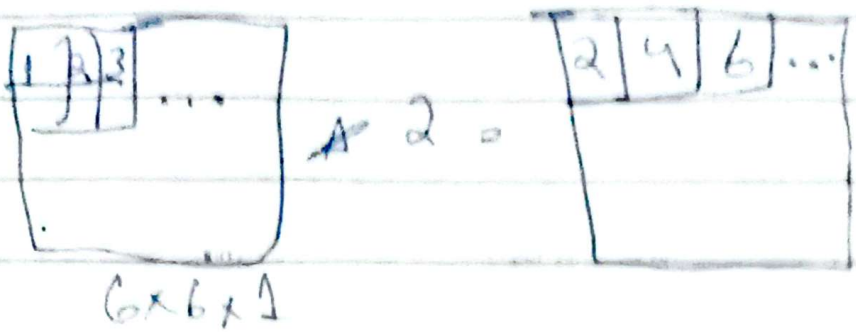
Why Resnets Work?



give more accuracy
not hurting performance

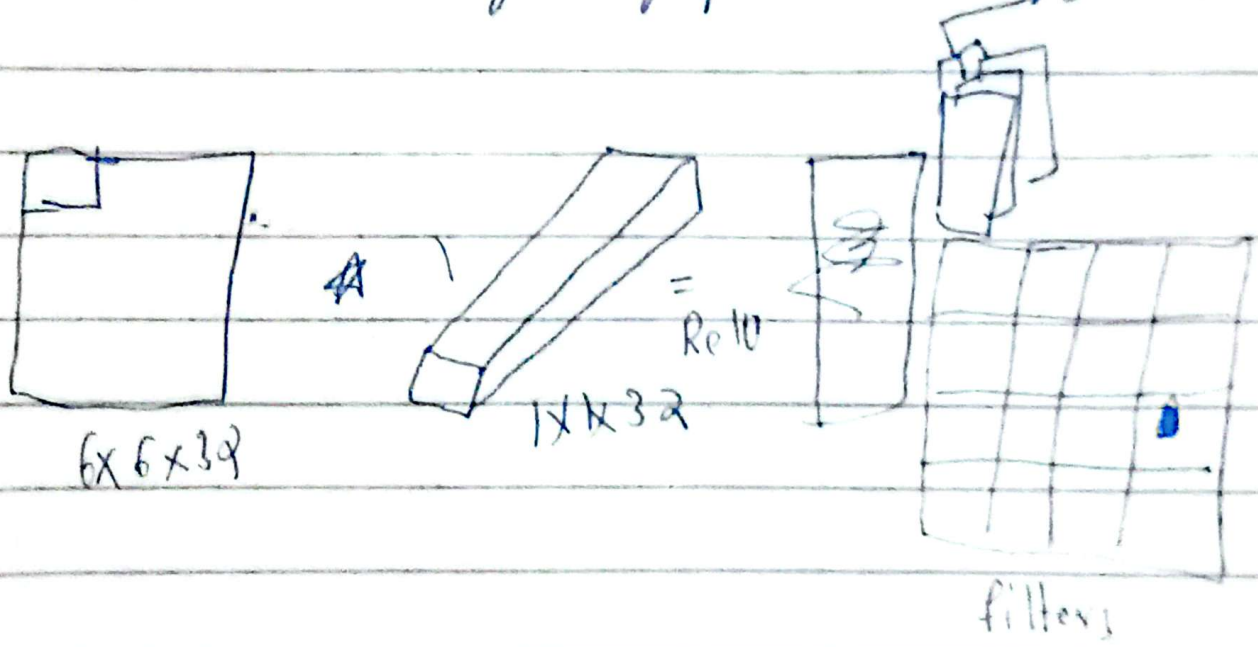
They are skip laye bcz the skip over one over one or more layers and is added directly

1×1 Convolution and Network in Network



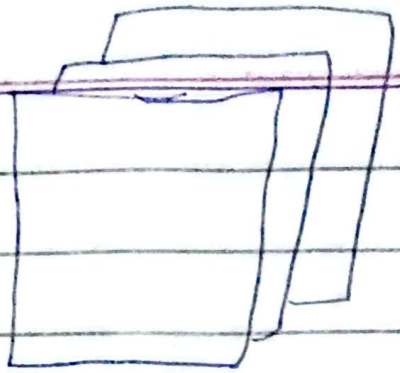
1×1 Convolution

- Filter size 1 height × 1 width
- It slides over the input feature map one spatial position at a time.
- It only mixes information across channels, don't across neighboring pixels.



channels = layers

$6 \times 6 \times 32 * 1 \times 1 \times 32 =$



6x6 x, filter,

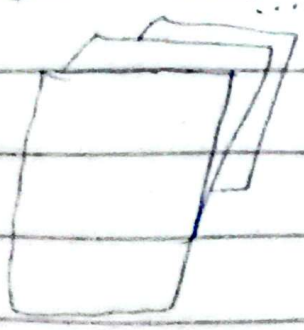
one create many ...

feature map

Like an example

We want to create $1 \times 1 \times 32$

32 filters from 192

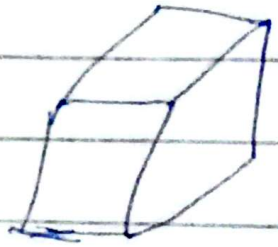


192

ReLU

Conv 1x1

32

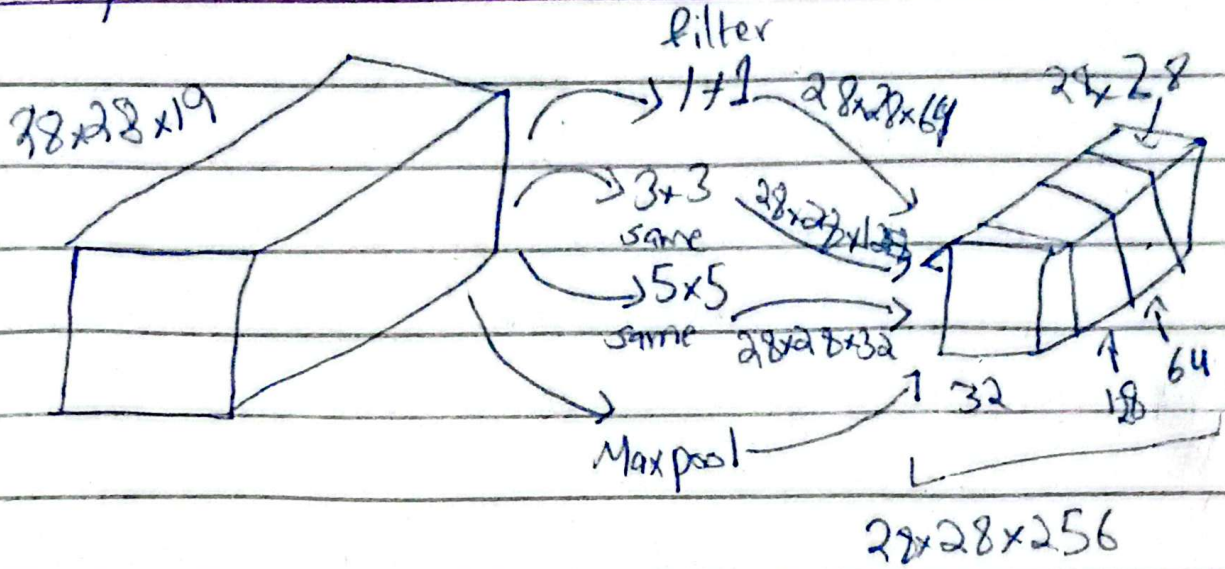


28x28x32

28x28x192

we created 32 filter features or channels

Inception Network Motivation



Motivation Befor 2014

- Small filters capture fine details but miss larger context.
- larger filters capture more context but are expensive and prone to overfit.

Idea:-

Why not use multiple filter and network deaide which is useful.

Why we use it.-

- To reduce computation cost with 1×1 as bottlenecks.
- To make network deeper and wider without exploding computation.

3) How it Works.

1x1 captures fine details, reduces dimension.

2) 1x1 conv → 3x3 con

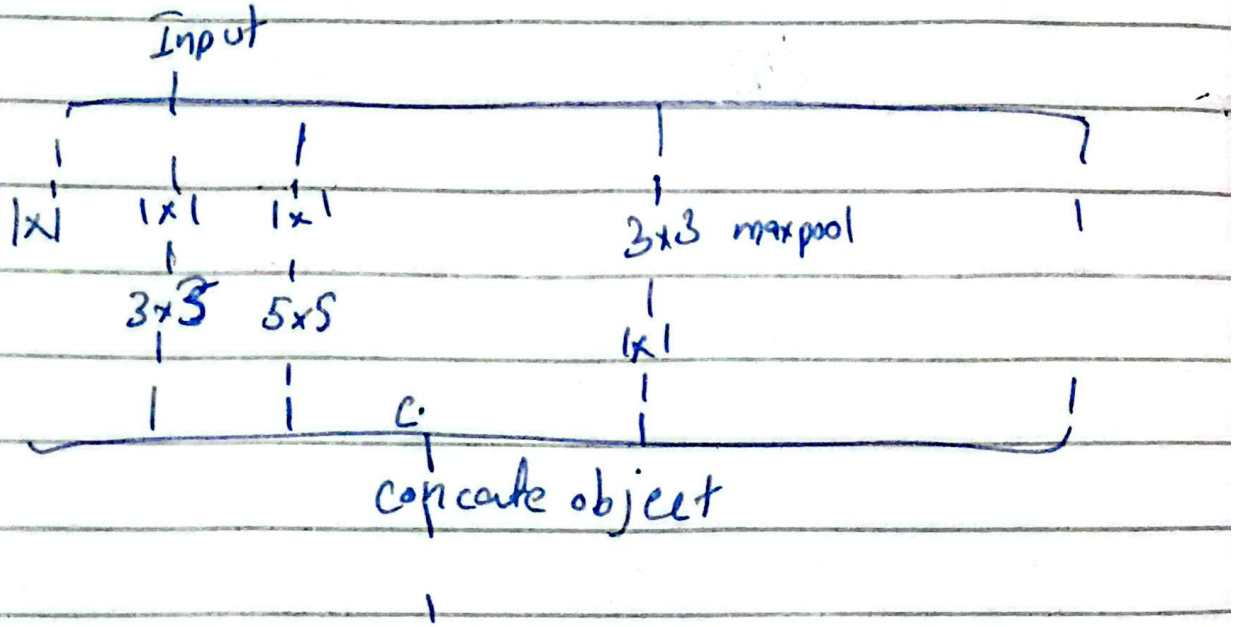
First reduce channel, then detect medium-scale patterns.

3) 1x1 conv → 5x5 conv

Reduce channel and detect large scale.

4) 3x3 max pooling → 1x1 conv

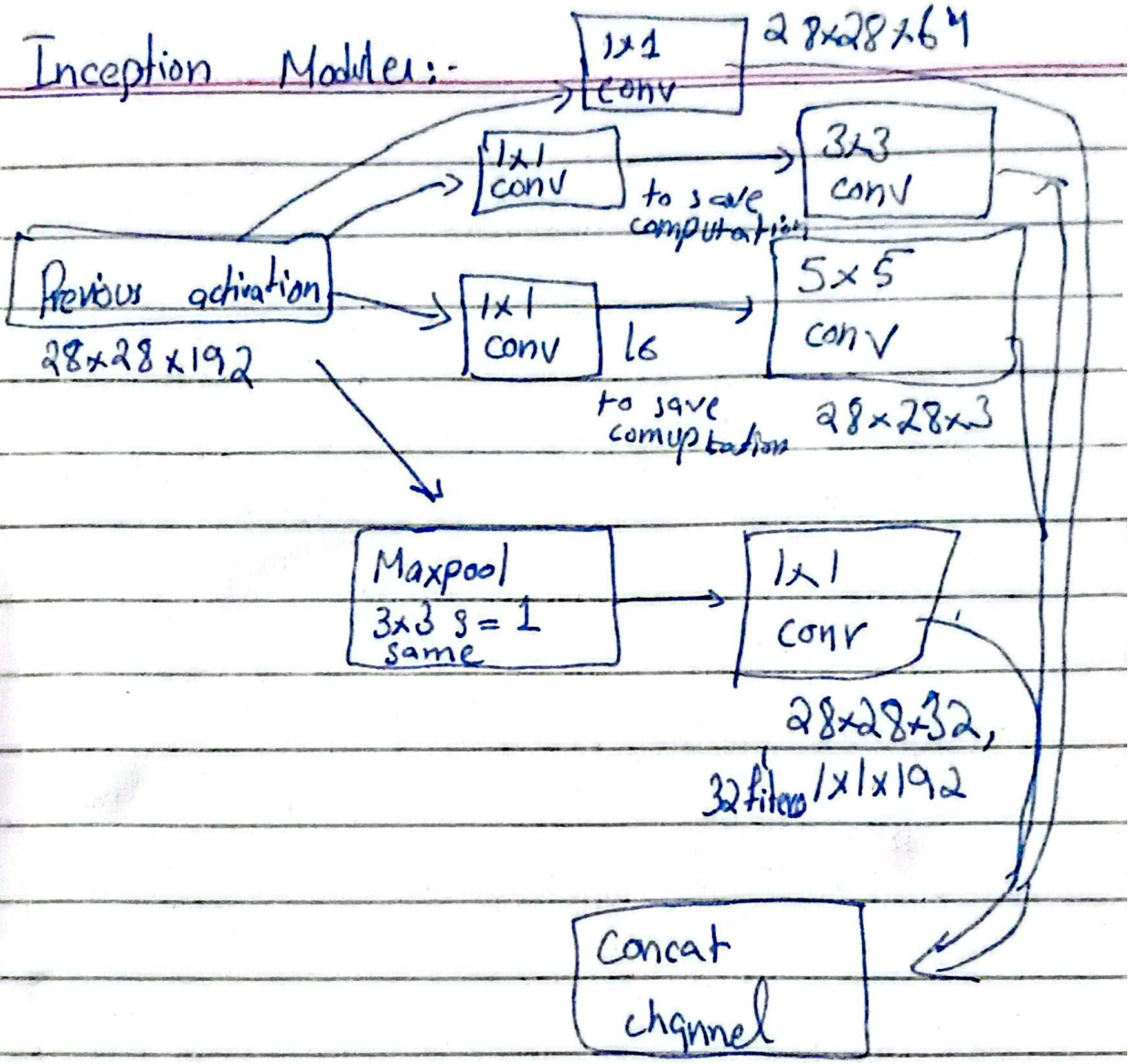
Capture background, then mixes channels



When Use:-

When we can't decide use 1x1, 3x3, 5x5 convolutions the we use Inception Network motivation

Inception Module:-



6×6×32 d=filters

w×h× filters

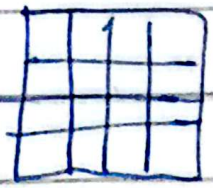
w×h×d

Mobilenets:-

⇒ efficient, can use even in mobiles

Normal Convolution = computation cost 2160

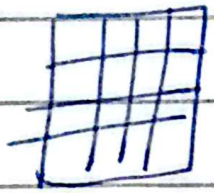
$6 \times 6 \times 3 \quad * \quad 3 \times 3 \times 3 = 4 \times 4$



Depthwise convolution (RGB)

red

$6 * 6 * 3 * \quad 3 \times 3 = 4 \times 4 \times 3$

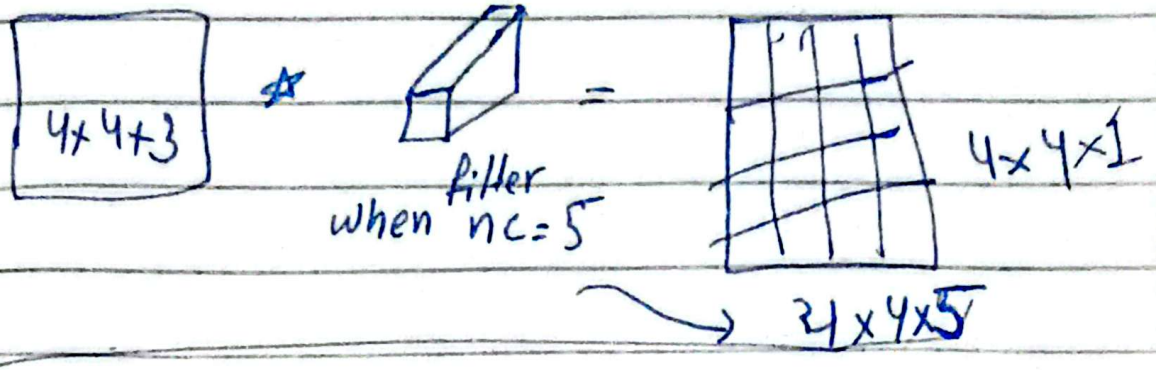


~~$6 \times 6 * 3 * \quad 3 \times 3$~~ Computation cost = 540

What happen

* this multiply single individual layers (3 RGB one by one) with a individual filter (red for red filter, blue with blue, green with green).
* This individual layer multiplicatiacation cause in reduce computation cost.
* This we can we this in mobiles.
* After this it again subject it to pointwise convolution

result ↓ further multiply to point wise



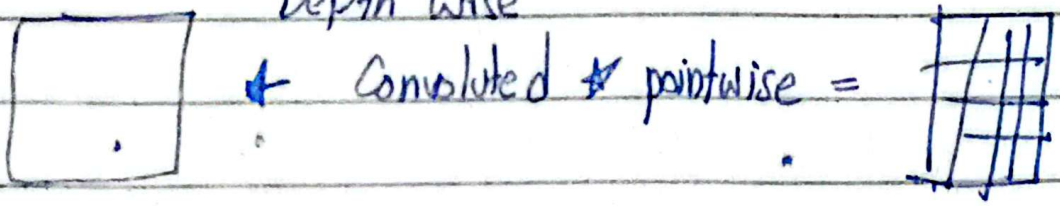
Paintwise

computation cost = filter param × filter positions × # filters
(no. of)

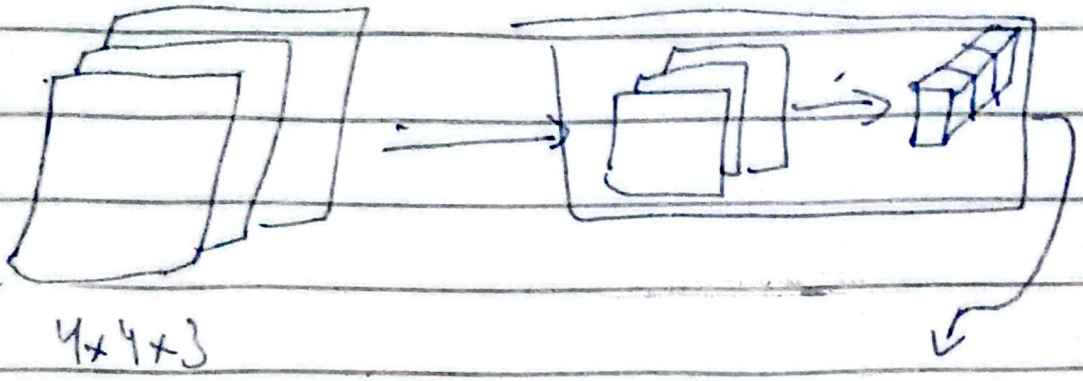
= 1x1x3 4x4 5

Computation = 240

Mobile Net use



MobileNet Architecture:-

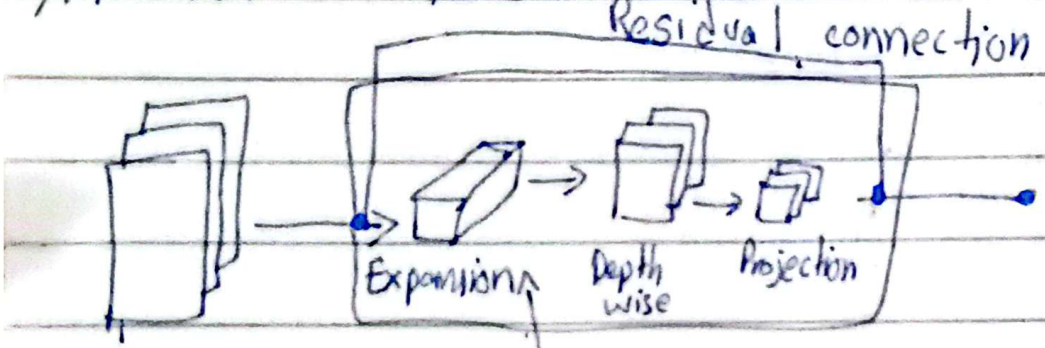


Pool → FC → Softmax.

Improvement in it.

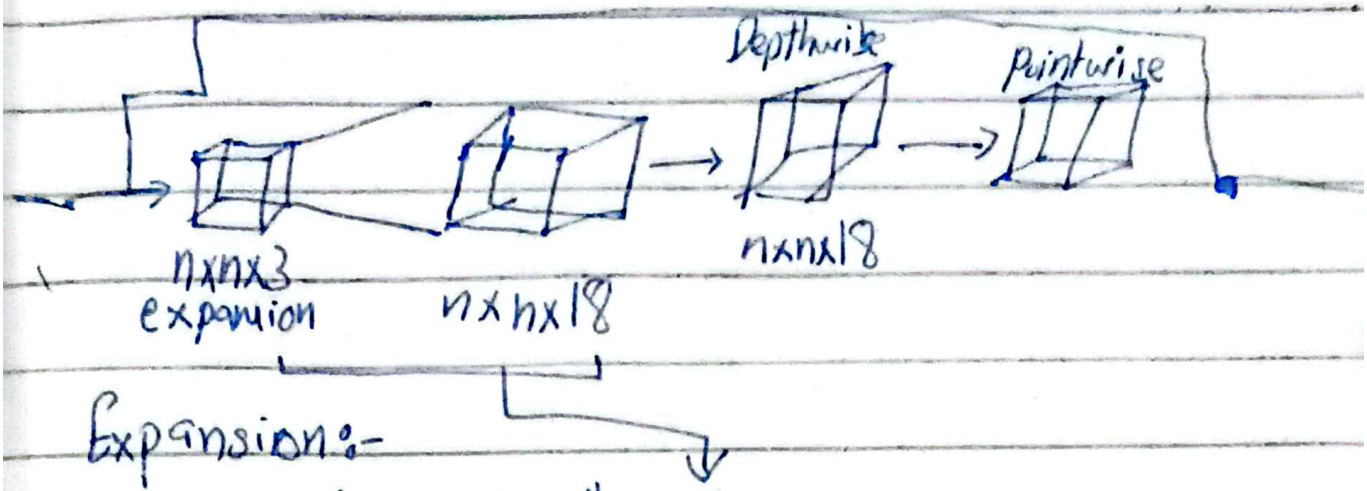
There are two improvement:

1) Addition of Residual Connection



2) Addition of expansion connection

Residual connection

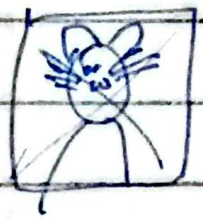


Expansion:-
increase the size

Use of depthwise and pointwise cause decrease in computation

Efficient.NET

~~Baseline~~



We can find on efficient ~~way~~ way by efficient net to up and down the (wxhxd) for a specific device.

Transfer learning

is a technique of taking a CNN that has already has been trained on a large dataset (like Imagenet with millions of images) and adapting it for a new, often smaller task

1) Feature extraction

Freeze most of all pre-trained CNN layers

DATA AUGMENTATION

This same as changing axis, scale size, Rotation, Mirroring is data augmentation.

Module 3 Object Detection

Object localization

localization is to detect where the car is

Landmarks

If we want to detect hand from the body we will train the machine by providing it the picture, cordinates and bounding boxes for its training e.g

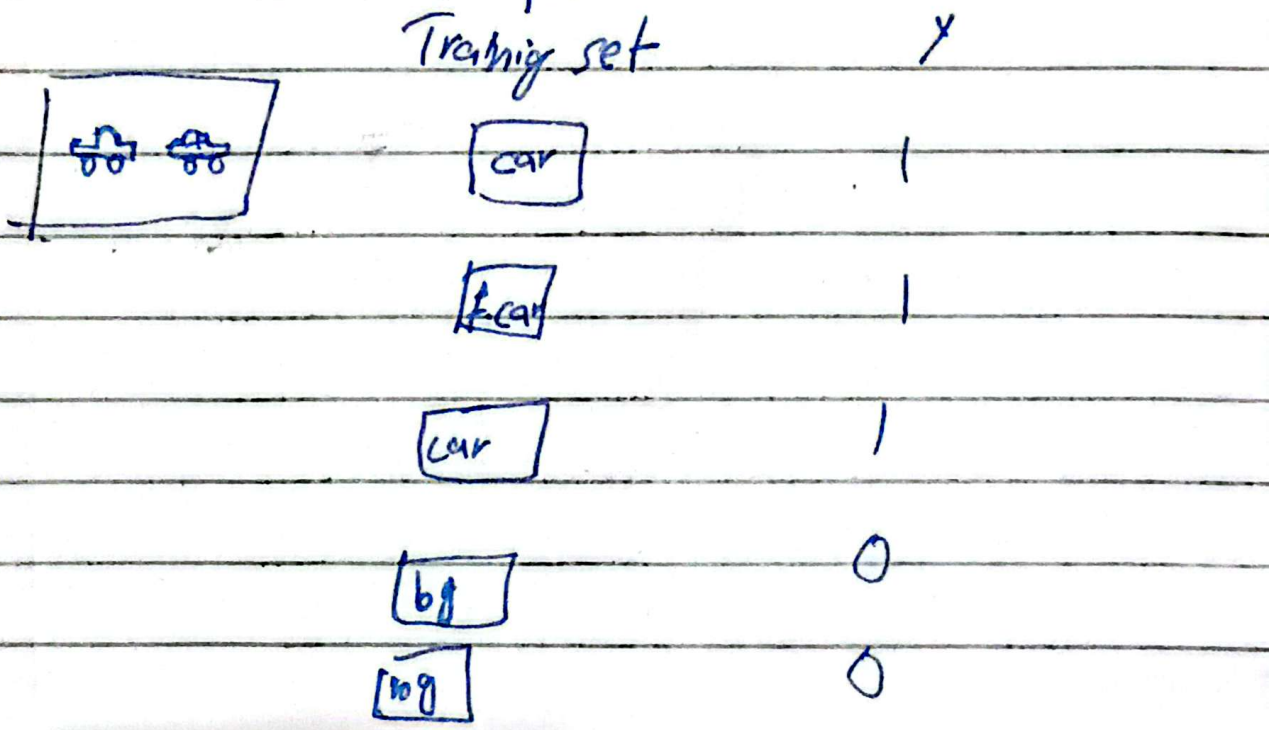
detect the eye

$L_1 \ldots, L_2$, for detecting a corner

$L_3 \ldots L_4$, for detect another corner

⋮

Car detection example



Running or sliding windows require computation but we have solution

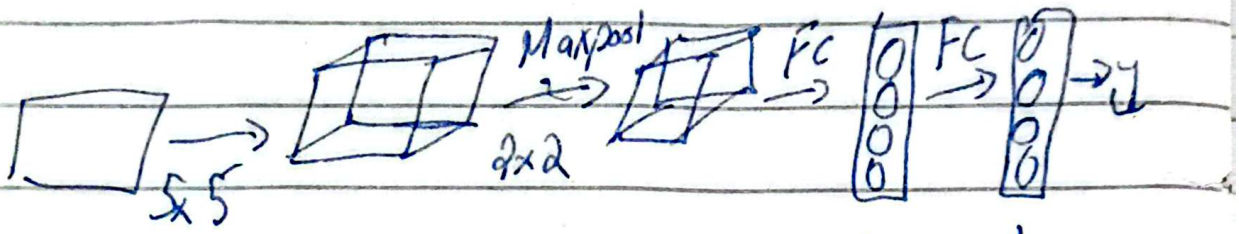
Solution:

I # Convolution Implementation of sliding windows

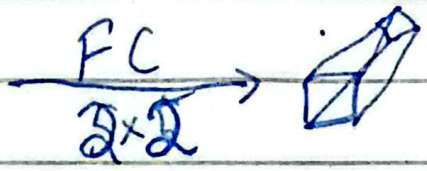
2×2×4

24×24

First Turning FC layers to convolution layer



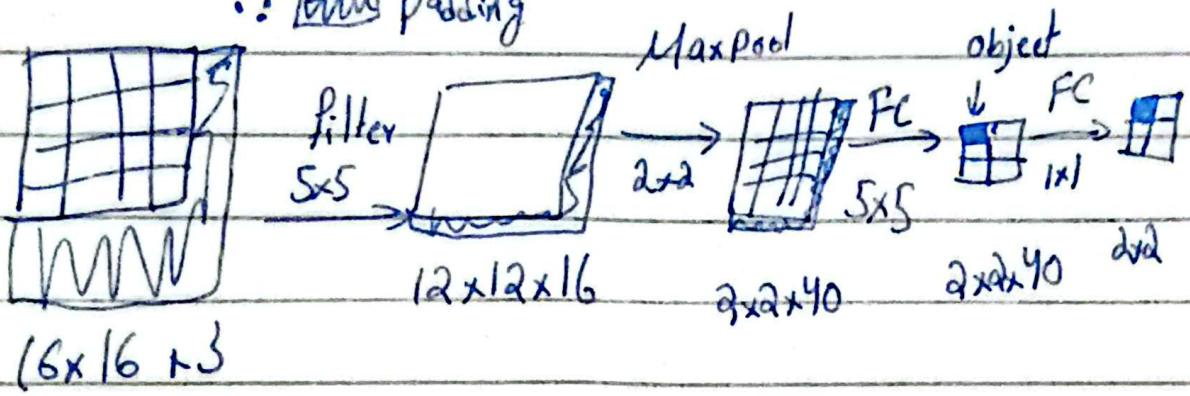
How to make them conv layer.



IMPLEMT of sliding windos

(FC) FC has been converted in convolution

∴ Padding



but this process is not too accurate.

It don't output correct bounding boxes.

Solution:-

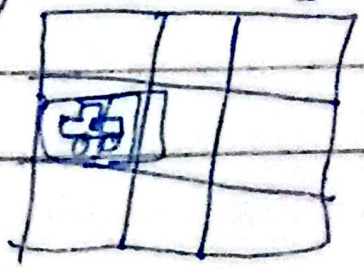
Banding box Predictions.

⇒ Yolo algorithem

⇒ It pointe the midpoint of taryet bounding box.

⇒ treat detection as a single regression problem rather than tunning classifier on many region.

⇒ Yolo allows each grid cell to predict multiple bounding boxes for different object shape

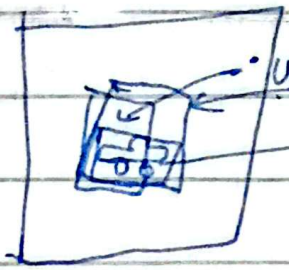


OD

How to check algorithem is working well

- Intersection over union

$$= \frac{\text{size of union}}{\text{size of intersection}} \quad \because \text{corect if } IOU \geq 0.5$$



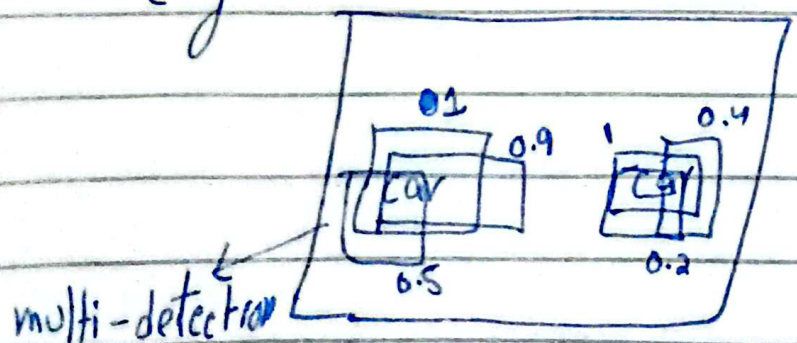
correct if $IOU \geq 0.5$

It show us how correct bounding box is, it should be 0.5, higher the number Greater accuracy is

IOU is the measure of overlap b/w two bounding boxes.

TA Non-max Suspension:- (Help to work even better)
(It pervents multiple detection of same object.

e.g



Non-max suppression pervent this.

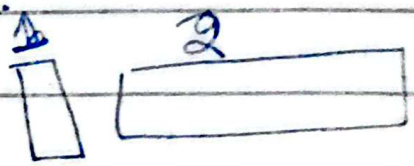
(as 1)

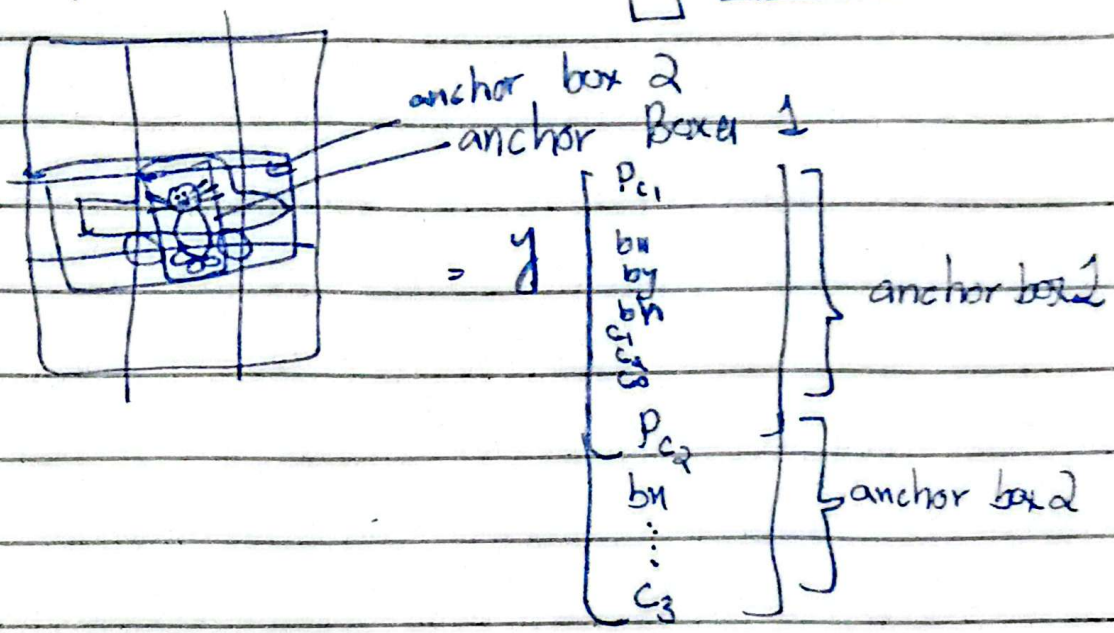
This cause to take max probablity and other boxes are suppressed with less probablity.

T# Anchor Boxes:- OD (object detection)

One of problems is that ~~the~~ one grid detect only one boxes.

e.g





Previous with one box or object

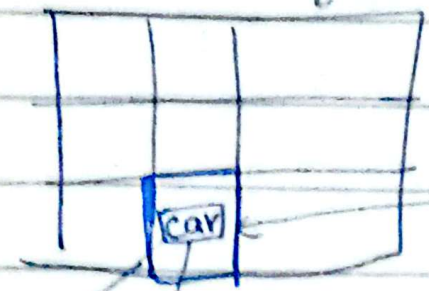
$3 \times 3 \times 8$

now ← 2 anchor boxes.

$3 \times 3 \times 2 \times 8$

Put all the things (IOU, Anchor, Non-max suppression) in Yolo

Yolo algorithem



no anchor grid as only car is present

IOU 0.5

greater probability suspension

model score

The Region proposal cell (often less used)

* This allows the sliding window where it find the target rather than the sliding over the whole image.
* This algoritham is slow

Faster R-CNN:

Use convolution network to propose but slower than Yolo algorithm

Semantic and Segmentation with U-Net

Semantic segmetation

is a computer vision task (car-driving) where the goal is to assign a class label to every single pixel in an image.

Search object vs Sematic segmetation

Useful

Chest X-Ray, Brain MRI

~~It~~ is

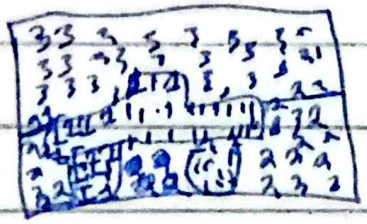
Working :

It assigned the pixel of specific thing (like car, rod, building) with number

1 for car

2 = rod

3 = Building



and also give color to ever pixel

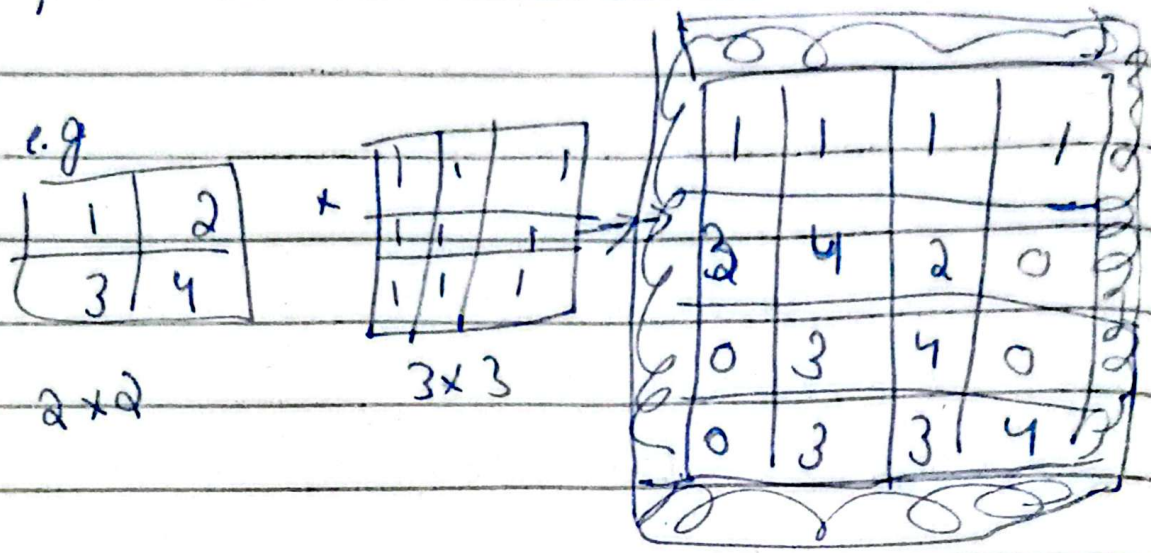
red = 1

Blue = 3 = Building

2 = Black = road.

Transpose convolution:

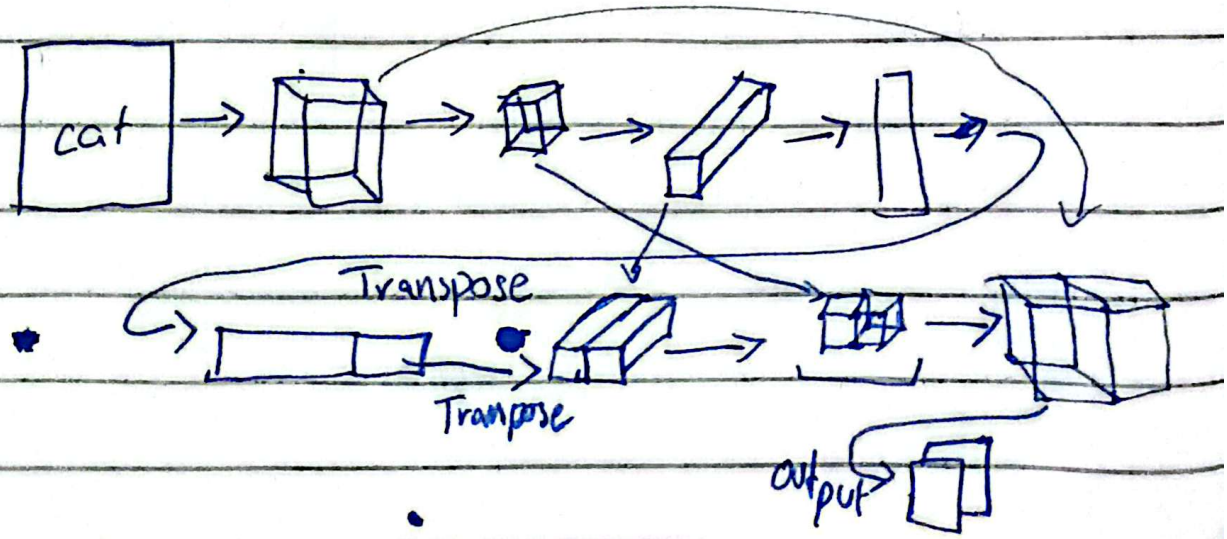
is a way to generate large output from the smaller one



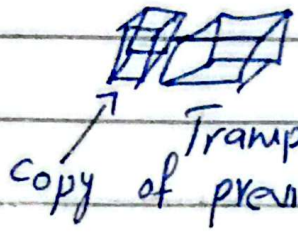
padding = P=1 , stride s=2

When two box overlap we ~~multiply~~ add then

U-Net Architecture intuition:.



After taking transpose it also merge the previous copy in UNet



Week #4

Face Recognition Vs face verification

Verification:-

- Input Image, name/ID.
- Output whether the input image is that of the person claimed person (Detect + comparison)

Recognization.

- Has a database of K persons number of
- Get Input
- Output ID if the image is any of K persons.

Image + compare from dbt get ID

One shot learning / algorithm recognize in one view

☆ Like ability of algorithem to recognize by any image by getting one image of that person.

☆ Recognize diffirent axis of face angle by trained on only one image of that face.

If (image 1, Img 2) $\leq \tau$
$> \tau$

Siamese network.

is a type of neural network architecture that contains two or more identical subnetworks that share the same weights and parameters.

a) Each input is passed through the same neural network. This network extracts feature vectors from both inputs.

3) The network calculates a distance or similarity score between the two vectors.

4) If the inputs are different, the distance

should be large.

Triplet loss.

$$\ell(A,P,N) = \max(\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha, 0)$$

Def:-

Triplet loss function is the loss function mainly used in metric learning and tasks like face recognition, image retreival, or signature verification.

Work:-

It uses three inputs.

1) Anchor
2) Positive (example similar to anchor)
3) Negative (example not similar to anchor)

Mathematical formula:-

$$\text{Loss} = \max(0, d(f(A), f(P)) - d(f(A), f(N)) + \alpha)$$

Example:-

Suppose: $d(A,P) = 0.5$, $d(A,N) = 0.9$

$\alpha = 0.3$

$L = \max(0, 0.5 - 0.9 + 0.3) = \max(0, -0.1) = 0$

Choose triplets that are hard to train.

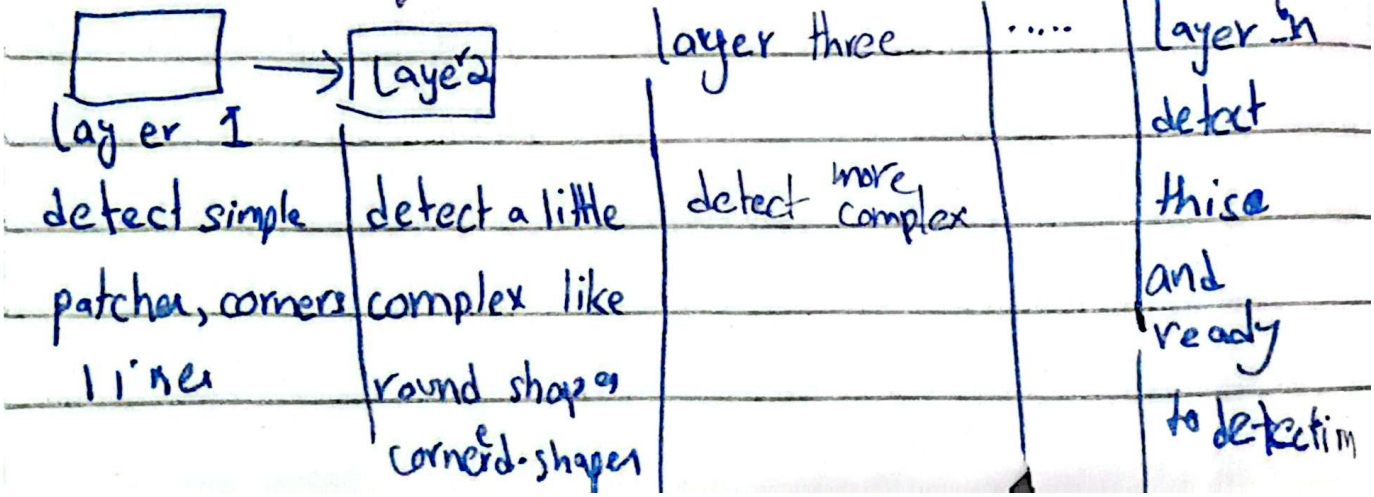
T# Face verification and binary classification.

$$\hat{y} = \sigma\left(\sum_{k=1}^{128} |f(x^{(i)})_k - f(x^{(j)})_k| + b\right)$$

$$\text{chi square} = \frac{(f(x^{(i)})_k - f(x^{(j)})_k)^2}{f(x^{(i)})_k + f(x^{(j)})_k}$$

Binary classification is the more general ml task where the system classifies input data into one of two categories or classes.
Input: A single data sample (Two faces)
Output: One of two classes (Yes or No)

T# Neural style transfer



Deep Convolutional Neural Networks.

These use convolution layers to automatically and adoptively learn spatial hierarchies of features — from simple edges in early layers to complex objects in deeper layer.

1# Neural style transfer cost function - of Generated image

$$J(G) = \alpha J_{content}(C, G) + \beta J_{style}(S, G)$$

2) Use gradient decent to minimize

$$G = G - \frac{\partial}{\partial G} J(G)$$

1# How to define content cost function

⇒ $J(G) = \alpha J_{content}(C, G) + \beta J_{style}(S, G)$

Say you use hidden layer l to compute content cost.

⇒ Use pretrained ConvNet

⇒ Let $a^{[l][C]}$ and $a^{[l][G]}$ be the activation of layer l on the images.

If $a^{[l][C]}$ and $a^{[l][G]}$ are similar than they similar content.

Hence, this is use to detect different content

$$J_{content}(G,G) = \frac{1}{2} \| a^{[l][C]} - a^{[l][G]} \|^2$$

$$(G_S - G_G)^2$$

instead of difference :

$$(G_S - G_G)$$

The style cost function should be:

$$J^{[l]}_{style}(S,G) = \frac{1}{(2n^{[l]}_H n^{[l]}_W n^{[l]}_C)^2} \sum_k \sum_{k'} \left(G^{[l][S]}_{kk'} - G^{[l][G]}_{kk'} \right)^2$$

If $a^{[l][c]}$ and $a^{[l][g]}$ are similar than they similar content.

Hence, this is use to detect different

$$J_{content}(C,G) = \frac{1}{2} \| a^{[l](c)} - a^{[l][g]} \|^2$$

$$(G_S - G_G)^2$$

instead of difference :

$$(G_S - G_G)$$

The style cost function should be:

$$J^{[l]}_{style}(S,G) = \frac{1}{n_H \cdots}$$

$$= \frac{1}{(2 n_H^{[l]} n_W^{[l]} n_C^{[l]})^2} \sum_k \sum_{k'} \left(G^{[l](S)}_{kk'} - G^{[l](G)}_{kk'} \right)^2$$

Style cost function

Style:

as correlation between activations across channels. (corr) between style of patches

Style matrix

Let $a^{[l]}_{i,j,k}$ = activation at (i,j,k). $G^{[l]}$ $n_c^{[l]} \times n_c^{[l]}$

H w c

> Note: Neural style transfer use this style matrix to generate new images that combine the content of one image with the style of another.

> Style Matrix: Encodes style features for style transfer.

capture style of S

$$G^{[l][S]}_{kk'} = \sum_{i=1}^{n_H^{[l]}} \sum_{j=1}^{n_w^{[l]}} a^{[l](S)}_{ijk} \, a^{[l](S)}_{ijk'}$$

Capture style of S

$$G^{[l](S)}_{kk'} = \sum_{i=1}^{n_H^{[l]}} \sum_{j=1}^{n_w^{[l]}} a^{[l][G]}_{ijk} \, a^{[l][G]}_{ijk}$$

$$J^{[l]}_{style}(S,G) = \| G^{[l][S]} - G^{[l](G)} \|_F^2$$

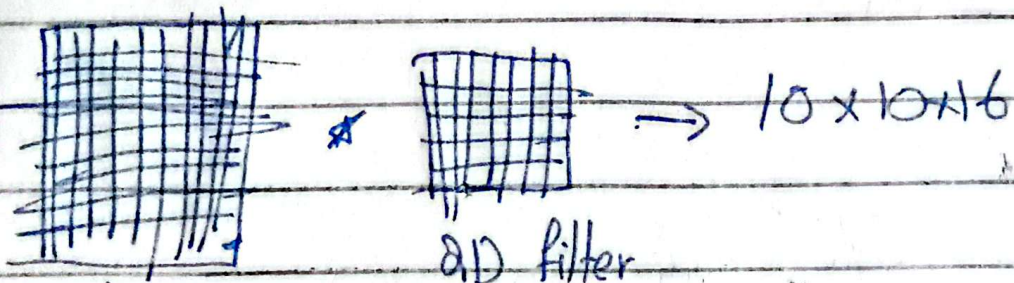
$$= \frac{1}{(2n_H^{[l]} n_w^{[l]} n_c^{[l]})^2} \sum_k \sum_{k'} \left(G_{kk'}^{[l][S]} - G_{kk'}^{[l](G)} \right)^2$$

Overall style cost function

$$J_{style}(S,G) = \sum_l \lambda^{[l]} J_{style}^{[l]}(S,G)$$

$$J(G) = \alpha J_{content}(C,G) + \beta J_{style}(S,G)$$

1D, 2D and 3D Images.



2D filter

2D

When we use ecg (1D) Data

14 . 5

14*1 * 5*1
→ 10*16*5*16
→ 6*32

What about 3D data
CT scan, organs scans

→ 10*10*10

Sequence model:-

Sequence model is one which trained on sequential Model.

* DNA sequence
* Speech Recognition
* Music Generation.