FACIAL EXPRESSION RECOGNITION & FACE DETECTION USING D-CNN – A DEEP VISION

Dissertation submitted in partial fulfilment of the requirements for the Degree of

MASTER OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

PUNEET SAMNANI

11611827

Supervisor

Mrs. RICHA JAIN

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

Month- November Year – 2017

LOVELY PROFESSIONAL UNIVERSITY

TOPIC APPROVAL PERFORMA

School of Computer Science and Engineering

Program : P172::M.Tech. (Computer Science and Engineering) [Full Time]

COURSE CODE : CSE548 **REGULAR/BACKLOG :** Regular **GROUP NUMBER :** CSERGD0054

Supervisor Name : Richa Jain **UID :** 17688 **Designation :** Assistant Professor

Qualification : M TECH **Research Experience :** 5+ YEARS

SR.NO.	NAME OF STUDENT	REGISTRATION NO	BATCH	SECTION	CONTACT NUMBER
1	Puneet Samnani	11611827	2016	K1637	9455152237

SPECIALIZATION AREA : Networking and Security **Supervisor Signature:**

PROPOSED TOPIC : Image Recognition System using Deep Learning - Introducing a new Deep Vision .

Qualitative Assessment of Proposed Topic by PAC		
Sr.No.	**Parameter**	**Rating (out of 10)**
1	Project Novelty: Potential of the project to create new knowledge	7.75
2	Project Feasibility: Project can be timely carried out in-house with low-cost and available resources in the University by the students.	7.50
3	Project Academic Inputs: Project topic is relevant and makes extensive use of academic inputs in UG program and serves as a culminating effort for core study area of the degree program.	7.00
4	Project Supervision: Project supervisor's is technically competent to guide students, resolve any issues, and impart necessary skills.	7.50
5	Social Applicability: Project work intends to solve a practical problem.	7.00
6	Future Scope: Project has potential to become basis of future research work, publication or patent.	7.00

PAC Committee Members		
PAC Member 1 Name: Prateek Agrawal	UID: 13714	Recommended (Y/N): Yes
PAC Member 2 Name: Deepak Prashar	UID: 13897	Recommended (Y/N): Yes
PAC Member 3 Name: Raj Karan Singh	UID: 14307	Recommended (Y/N): NA
PAC Member 4 Name: Pushpendra Kumar Pateriya	UID: 14623	Recommended (Y/N): Yes
PAC Member 5 Name: Sawal Tandon	UID: 14770	Recommended (Y/N): NA
PAC Member 6 Name: Aditya Khamparia	UID: 17862	Recommended (Y/N): Yes
PAC Member 7 Name: Anupinder Singh	UID: 19385	Recommended (Y/N): NA
DAA Nominee Name: Kuldeep Kumar Kushwaha	UID: 17118	Recommended (Y/N): NA

Final Topic Approved by PAC: **Image Recognition System using Deep Learning - Introducing a new Deep Vision .**

Overall Remarks: Approved

PAC CHAIRPERSON Name: 11024::Amandeep Nagpal **Approval Date:** 30 Nov 2017

11/30/2017 3:00:36 PM

ABSTRACT

Facial emotion recognition is one of the most important cognitive functions that our brain performs quite efficiently. State of the art facial emotion recognition techniques are mostly performance driven and do not consider the cognitive relevance of the model. This project is an attempt to look at the task of emotion recognition using deep belief networks which is cognitively very appealing and at the same has been shown to perform very well for digit recognition . We look at the effects of varying number of hidden layers and hidden units on the performance of the model and attempt to develop important insights into the features learnt by the model. Also we observe that as found various psychological findings our model finds lower spatial frequency more useful for recognizing facial expressions than higher spatial frequency data.

DECLARATION

I hereby declare that the research work reported in the dissertation/dissertation proposal entitled " FACIAL EXPRESSION RECOGNITION & FACE DETECTION USING D-CNN – A DEEP VISION" in partial fulfilment of the requirement for the award of Degree for Master of Technology in Computer Science and Engineering at Lovely Professional University, Phagwara, Punjab is an authentic work carried out under supervision of my research supervisor Mrs. Richa Jain. I have not submitted this work elsewhere for any degree or diploma.

I understand that the work presented herewith is in direct compliance with Lovely Professional University's Policy on plagiarism, intellectual property rights, and highest standards of moral and ethical conduct. Therefore, to the best of my knowledge, the content of this dissertation represents authentic and honest research effort conducted, in its entirety, by me. I am fully responsible for the contents of my dissertation work.

Signature of Candidate

Puneet Samnani

R.No- 11611827

SUPERVISOR'S CERTIFICATE

This is to certify that the work reported in the M.Tech Dissertation/dissertation proposal entitled "**FACIAL EXPRESSION RECOGNITION & FACE DETECTION USING D-CNN – A DEEP VISION"**, submitted by **Puneet Samnani** at **Lovely Professional University, Phagwara, India** is a bonafide record of his / her original work carried out under my supervision. This work has not been submitted elsewhere for any other degree.

Signature of Supervisor

Mrs. Richa Jain

Date:

Counter Signed by:

1) **Concerned HOD:**
   HoD's Signature: _____

   HoD Name: _____

   Date: _____

2) **Neutral Examiners:**

   External Examiner

   Signature: _____

   Name: _____

   Affiliation: _____

   Date: _____

   Internal Examiner

   Signature: _____

   Name: _____

   Date: _____

ACKNOWLEDGEMENT

I have taken sincere efforts to make this Dissertation-II Report . However, it would not have been possible without the kind support and help of many individuals and the University. I would like to extend my sincere thanks to all of them.

I am highly indebted to **Mrs. Richa Jain** for her guidance and constant supervision as well as for providing me the necessary information regarding the Research Work & also for her support in completing the Report.

I would like to express my gratitude towards my parents for their encouragement which help me in completion of this Report.

TABLE OF CONTENTS

CONTENTS **PAGE NO.**

Table of Figures

LIST OF TABLES

CHAPTER1: INTRODUCTION

What do we see when we look at this image. Do we see a person looking at we or do we see a person looking to the right we can see that your brain is is struggling is struggling to adjust if we look to the right side of the image. Just look at the right border there which we'll see a person looking to the right. If we look at the left border of the image we'll see a person looking at we. And this just proves that what our brain is looking for when we see things is features depending on the features that it sees depending on the features that we process. You categorize things in certain ways. So when we look on the right side of the image we see certain features of a person looking to ride because they're closer to your center of focus and therefore your brain classifies as a person looking to the right. When we look to the left side of the image we see more features of a person looking at we and therefore your brain classifies it as such.

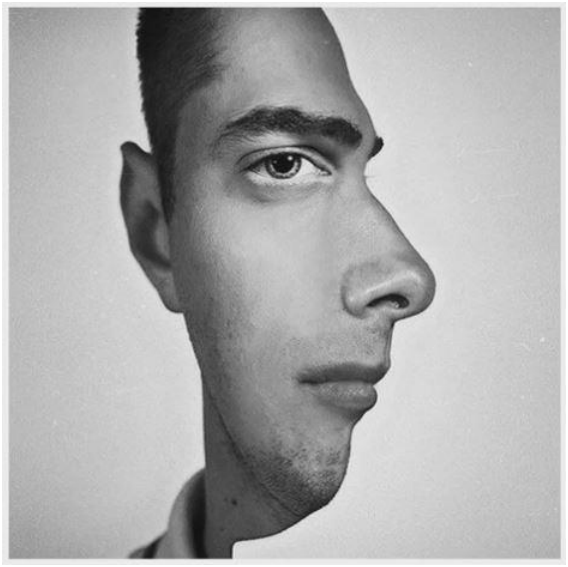


Figure 1.1 : Illusion Face 1

So let's have a look at another one. This is a very famous image. You probably have already seen it But what we see here. So some people will say that they see a young lady wearing a dress looking away. Some people say they see an old lady wearing a scarf on her head looking down. So I'm going to point this out and we'll see that will become very obvious so this is the face of the young lady looking away. She's looking into the distance as her coat. That's her hair that's her little feather in her hair and on the other hand. This is the head of the old lady looking down her nose her mouth her chin that's the scarf on her head and she's looking down. So as we can see two in one and depending on which features your brain picks up it will switch between classifying each image as one or the other.

Figure 1.2 : Illusion Face 2

The oldest one of these illusions recorded in the printed work is this one. It's the duck or the rabbit. So is this a duck or is this a rabbit.

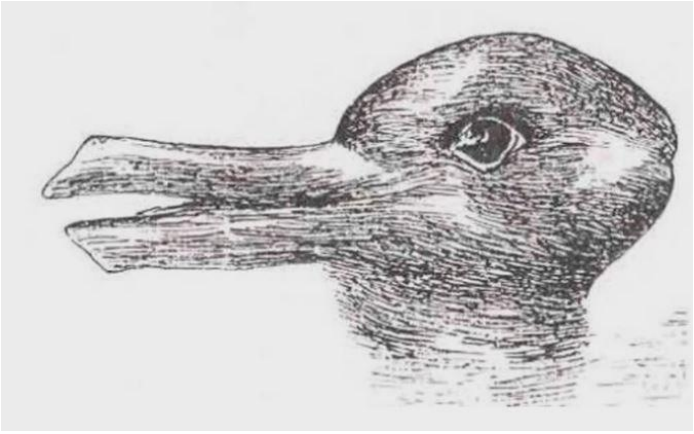


Figure 1.3 : Illusion Face 3

Another example. And now I'm going to show an image which will just for a second just look at it and see what emotions or what kind of experience visual experience we go through. So what do we see does we feel like a bit not dizzy but a little bit dazzled like your brain is trying to try and understand what it is what it is like it's trying to. Is jumping between her eyes up and down eyes and this is a classic example of when there are certain features where it could be this it could be that but your brain cannot decide. And because both seem plausible.

Figure 1.4 : Illusion Face 4

Yeah so basically all these examples illustrate to us how the brain works that it processes certain features on an image or on whatever we see in real life and it classifies that as. You probably have been in situations when we look over your shoulder quickly and we see something we think it's I don't know if it's like a a ball but it turns out to be a cat or we think it's a it's a car. Turns out to be a shadow or things like that that's because we don't have enough time to process those features or we don't have enough features to classify things as such.

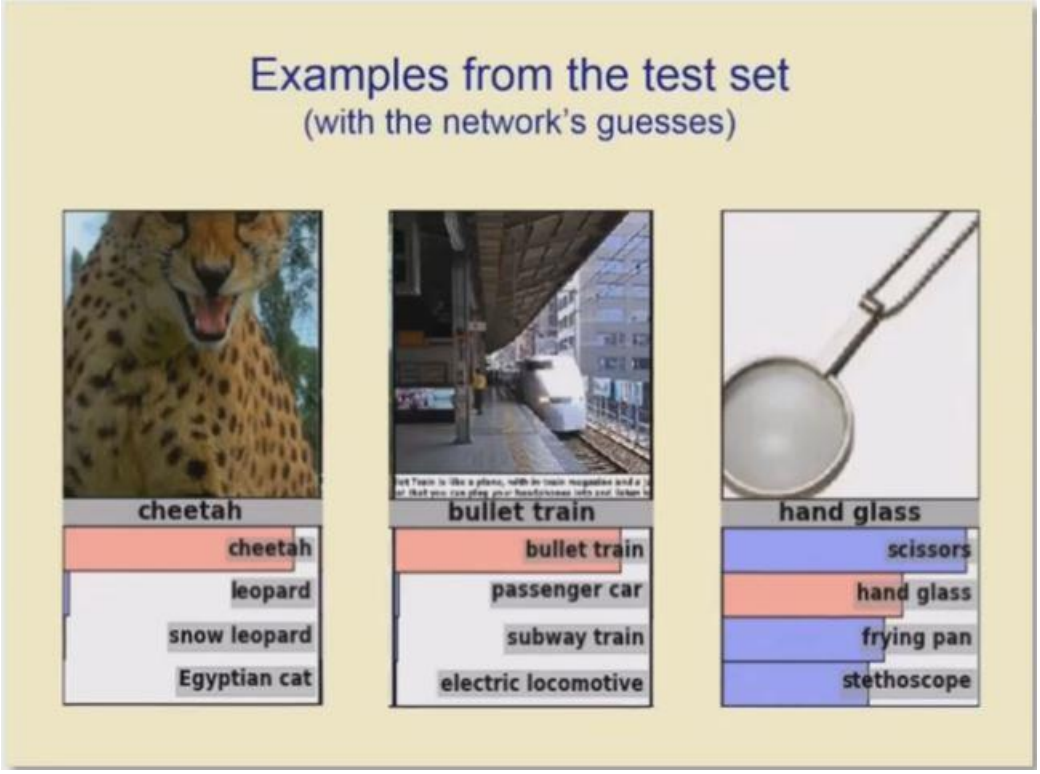


Figure 1.5 : Image Recognition Example

This is very interesting because what we're going to be doing with neural networks with have convolutional neural networks is very similar and we'll find that the way that computers are going to be processing images is going to be extremely similar to the way we are processing images so it's is very valuable to understand and just kind of remember these things that this is how we do it.

So here's something different. Here's an experiment an experiment done on computers on convolutional neural network so we're slowly moving now from humans to computers. So here we see three images and we're going to go through them with left to right and see how we would classify them and then see how they can be reclassified. So on the left what do we think this is. He probably said cheetah and we will be right.

And this is what the computer said so and the right right away right off the bat we're going to learn how to read these images because if we going to go deep into call convolutional neural networks no pun intended we're going to start learning more and more about and using them we'll see a lot of these. So and I've actually seen people read them incorrectly so here at the top Shida is what it actually is. So that's the actual correct label of the image that's what's the label of the images regardless of any processing. And the computer vision and then here are the guesses the top four or five sometimes guesses of the algorithm and they're given the probabilities so the computer said or the neural network said Chitta personal apparel or Egyptian cat can be one of the four. And cheetah has the highest vote. And throughout this part of the Course we understand what these votes mean and how they are derived. But for now it's pretty intuitive right. So it's a cheetah in reality and the neural network guessed right. It said with a hyper ability about like 95 99 percent. Then the second one. What do we think does it that is that it is a bullet train. And the neural network was able to distinguish between bullet train passenger car subway train electric locomotive.

Those are the top choice of course. It had many more options these neural networks learn to distinguish from not just four categories from dozens thousands of categories at the same time. So those are the four options that it picked. And so that's bullet train and its will. And so what did we think the last one is it very. There are a couple of options or it's not very clear what is it could be a frying pan could be a magnifying glass it could be even maybe a pair of scissors some might say while the neural network said it was a pair of scissors.

But we can see how we can go wrong here. First of all it's not a very clear image. And also we can see that the probabilities are not as clear here so the neural network was a bit confused a bit indecisive just as we are. So I said Scissors with the high probability but then it had hand-glass which it actually was with not so far away on second place and frying pan stethoscope. So basically here we can see that scissors was its first guess but the correct option was number two and that's why it's highlighted in red. So

there we go those That's what all the drugs are already capable of. And this is actually quite an old slide. This was several years ago. Now they're even better and we will see that from the practical application that we'll be coding together had lunch.

But now let's try this out a bit better. What convenance or convolutional neural networks actually are and why are they gained so much popularity. And they actually are gaining popularity so we can see here a Google Trends comparison I did just yesterday.

1.1 What is Convolutional Neural Networks (CNN)?

Here we can see that convention illusional neural networks are even taking over artificial neural networks so a massive increase. And this is going to keep going that way because it is a very important field that that is where all the things happen such as like self-driving cars.

How do they recognize people on the road how to recognize stop signs and things like that how do how does Facebook is Facebook able to tag images or people in images and not only just like remember previously years ago we had to tell people yourself then it would recognize faces we had to add the names.

And now it just recognizes the faces and adds the names at the same time. Well that is what convolutional neural networks are capable as being on Facebook. If Jeffrey Hinton is the godfather of artificial neural networks and deep learning then Yann Lecun is the grandfather of convolutional neural networks Lucun is a student of Jeffrey Hinton's and in fact here we can see them together. And Jeffrey Hinton now is pioneering de-planing at Google young. Is the director of Facebook artificial intelligence research and also a professor at NYU.

So we're slowly aware of this part of the core slowly we are building up this way. These names are this kind of picture of the profiles of the people who are driving this field and next in the next couple of pars will get to know about a few more and we'll have this whole Mafia as they call themselves or we can call them mafia or conspiracy of deep learning and we'll learn a bit more about how this whole field developed. Yeah it's just these are just some great great people.

And so Lecun back in the 80s and the 90s made significant contributions to the field of convolutional neural networks. And as we'll see throughout this course has been able to develop or help the world develop something so extremely powerful.

So moving on to how can illusional neural networks work. You have an input it's very simple it's very straightforward so they have an input image. It goes through the can illusional neural network and we have an label so it classifies that image as something

like has a Cheeto or a bullet train or something else. Now kind of like going into a bit more detail .

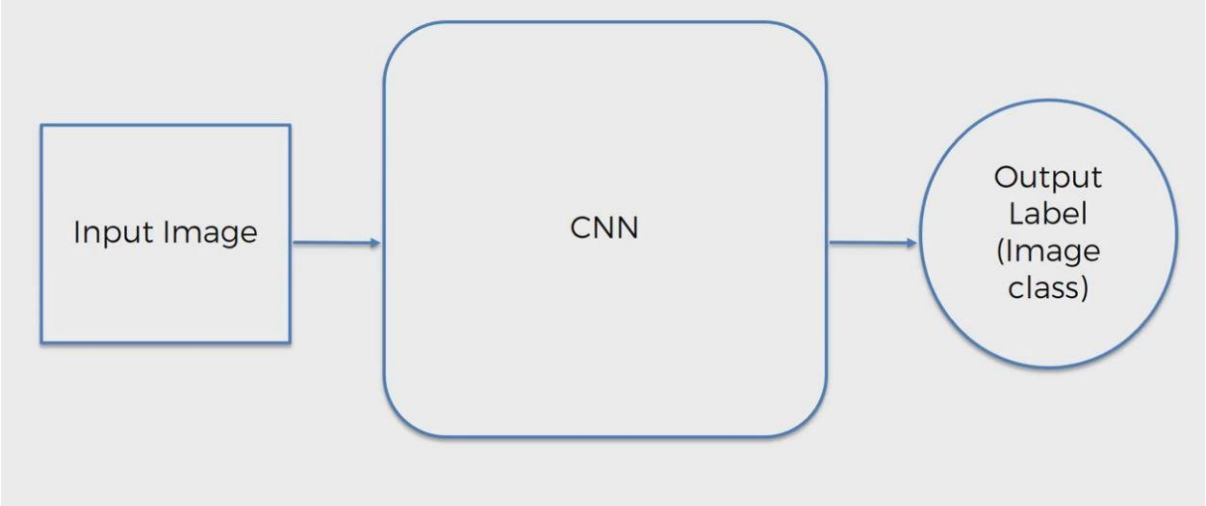


Figure 1.6 : Convolutional Neural Network - CNN

1.1.1 Emotion Detection using CNN

For instance we can officer neroli has been trained up on on certain images on certain classified images or categorized images prior there been higher prior. After that we can give it let's say a neural network has been trained up to recognize facial expressions and motions we can give it a face of a smiling person not just a face like a drawing of a face like this but actual face of a person smiling.

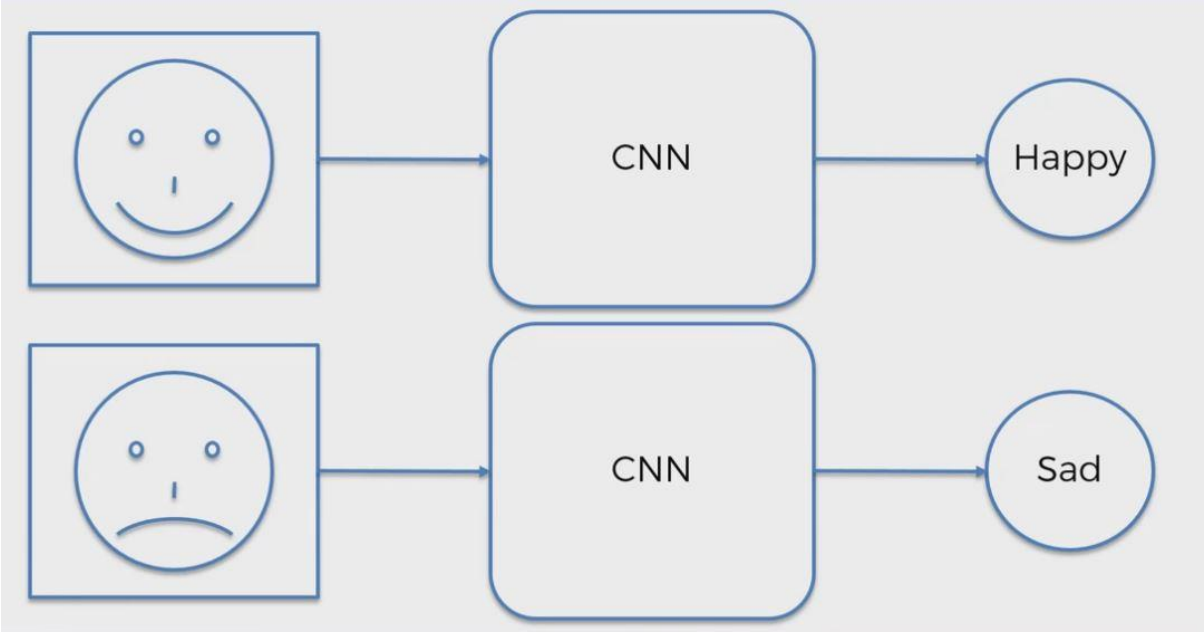


Figure 1.7 : Emotion Identification through CNN

And I'll tell we that that person is happy and we can get a face of a person that's frowning. I'll tell we that the person is sad. He can recognize these emotions and as we can see that's already very powerful in terms of so many different implications just this one example we can think of right away and in both cases Ill give we a operability so it won't say we know we're 100 percent the person's happy or sad. It'll be 99 or 98 or maybe 80 percent when it's unclear of what's going on and just like we are right sometimes we can mistake things for what they're not.

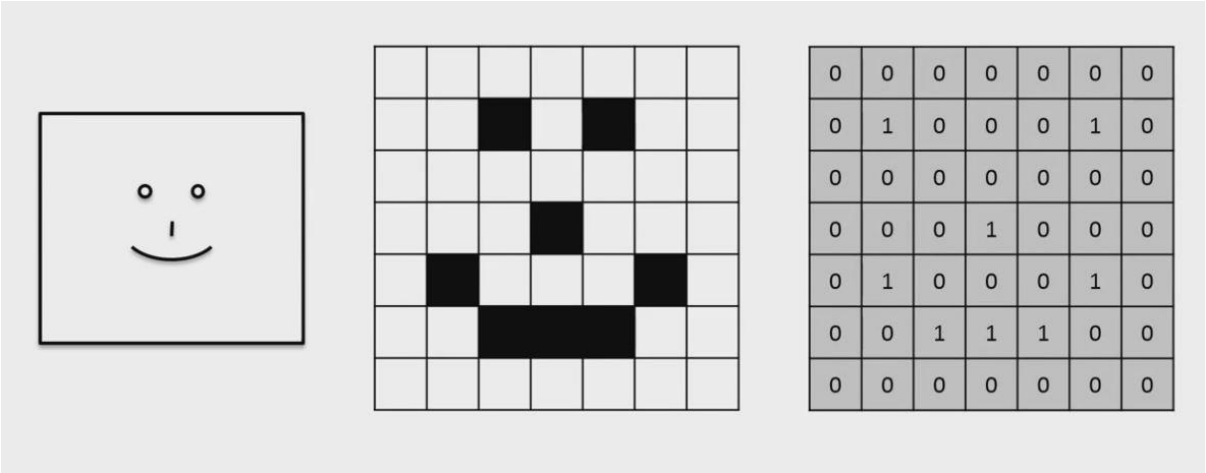


Figure 1.8 : Image to Binary Conversion

Or sometimes we can sometimes it's just not clear if the person is smiling or frowning or if it's if it's a dog or a cat or if it's a train or a bullet train. All right sometimes we don't have it we haven't seen enough features in all goes down to features because that's how we process visual information .

1.1.2 Converting Images into Binary

So but how does a neural network housing neural network able to recognize these features. Well it all starts at the very basic level we have. Let's say we have an image we have two images one is black and white image of two by two pixels and one is a color image of two by two pixels while neural networks leverage the fact that the black and white image is a two dimensional array.

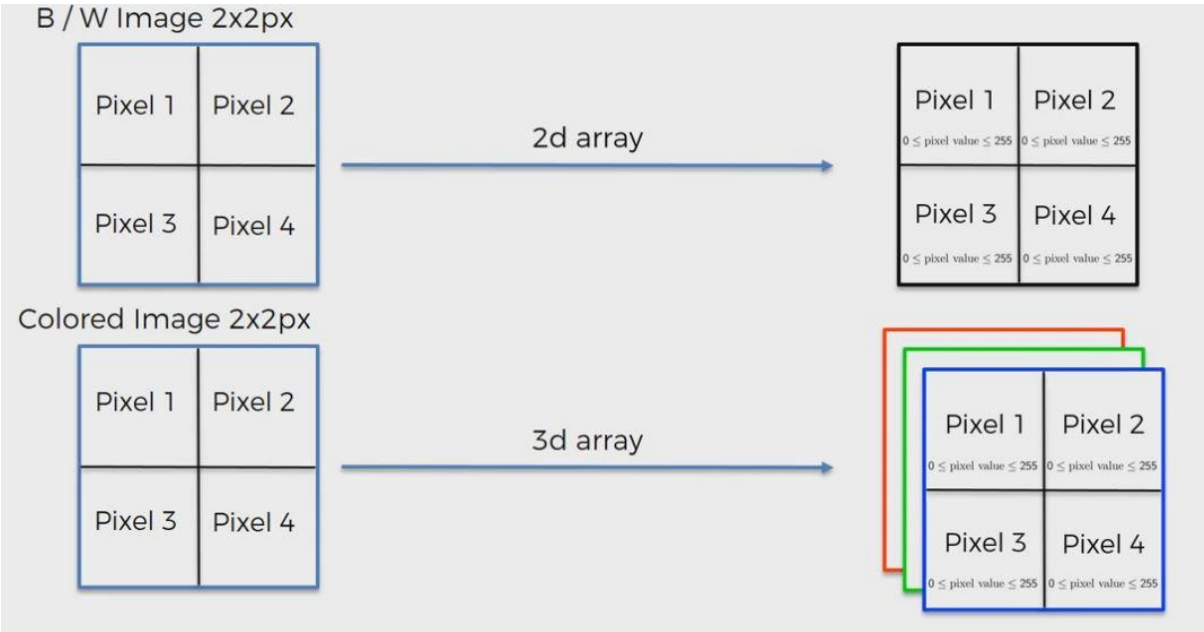


Figure 1.9 : Pixel Formation in 2D and 3D

So the way we see it right now on the left is just the visual representation. I suppose some kind of picture. And for simplicity's sake it's just a two way to picture but in computer terms it's actually a two dimensional array with every single one of those pixels having a value between 0 and 55.

So that's eight bits of information to the two to the power of eight is 256. So therefore the values from 0 to 255 and that's intensity of the color. And in this case the color white so 0 will be a completely black pixel. 255 will be a completely white pixel and between them we have the grayscale range of possible options for this pixel. And based on that information computers are able to then work with the image and that's kind of like the starting point that any image is actually has a digital representation has a digital form.

And those are just basically ones and zeros that form a number 0 to 255 for every single pixel and that's what the computer works with. It doesn't actually work with we know colors or anything it works with the ones and zeros at the end of the day. That's as kind of like the foundation of it all. And in a color image it's actually a three dimensional array. You've got blue pixel blue Larry Green and the red glare and arrows and that sense for RGV red green blue. And each one of those colors has its own intensity. So basically a pixel has three three values assigned to it.

Each one of them is between 0 and 256 255. And therefore we can find out what's this image what color exactly this pixel is. By combining those three values and again computers are going to be working with that. So that's the foundation of it all that's the red channel the green channel the blue channel. And finally let's have a look at for instance an example of a very trivial example of a smiling face. In computer terms.

If we just really simplify things instead of having from 0 to 255 and having those values just so that we can understand things better and really grasp the concepts we're going to say zero is is white one is black. So we're just going to simplify things to the extreme and we will see that that image can be represented like that. So the reason why we've brought this up is because we go into all of our intuitions Stroh's we get to structure an image is like this which is very simple but at the same time then all those concepts can translate back to the 0 2 256 range of values and everything applies the same way there.

CHAPTER 2 : REVIEW OF LITERATURE

We can divide our Literature survey into 3 divisions : Face Detection , Feature Extraction , Facial Expression Classification .

2.1 Face Detection

2.1.1 AlexNet(2012)

[35] The one that began everything (Though some may state that Yann LeCun's paper in 1998 was the genuine spearheading production). This paper, titled "ImageNet Classification with Deep Convolutional Networks", has been refered to a sum of 6,184 times and is broadly viewed as a standout amongst the most compelling productions in the field. Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton made a "substantial, profound convolutional neural system" that was utilized to win the 2012 ILSVRC (ImageNet Large-Scale Visual Recognition Challenge). For those that aren't natural, this opposition can be thought of as the yearly Olympics of Computervision, where groups from over the world contend to see who has the best Computer vision demonstrate for undertakings, for example, order, limitation, identification, and that's just the beginning. 2012 denoted the primary year where a CNN was utilized to accomplish a best 5 test mistake rate of 15.4% (Top 5 blunder is the rate at which, given a picture, the model does not yield the right mark with its main 5 forecasts). The following best passage accomplished a blunder of 26.2%, which was an astonishing change that basically stunned the Computer vision group. Safe to state, CNNs moved toward becoming commonly recognized names in the opposition from that point on out.

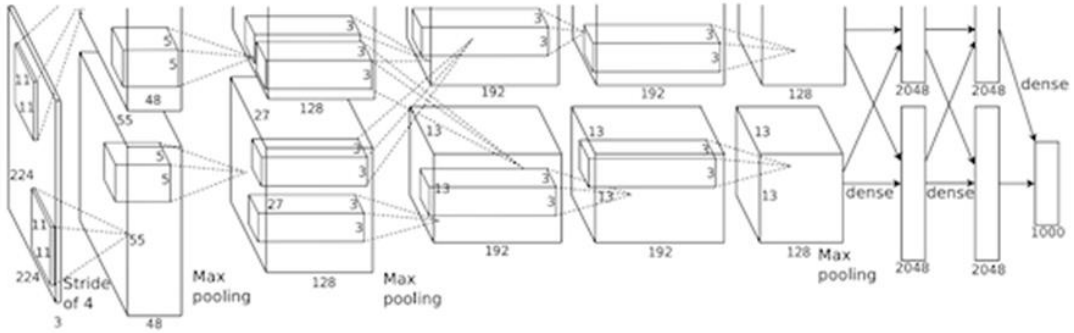


Figure 2.1 : AlexNet

In the paper, the gathering talked about the engineering of the system (which was called AlexNet). They utilized a generally basic design, contrasted with current structures. The system was comprised of 5 conv layers, max-pooling layers, dropout

layers, and 3 completely associated layers. The system they outlined was utilized for arrangement with 1000 conceivable classifications.

Trained the system on ImageNet information, which contained more than 15 million commented on pictures from a sum of more than 22,000 classifications. It used ReLU for the nonlinearity capacities (Found to diminish preparing time as ReLUs are a few times speedier than the customary tanh work). It used information expansion methods that comprised of picture interpretations, flat reflections, and fix extractions. It Implemented dropout layers with a specific end goal to battle the issue of overfitting to the preparation information. It trained the model utilizing group stochastic angle plunge, with particular esteems for energy and weight rot. It was trained on two GTX 580 GPUs for five to six days.

The neural system created by Krizhevsky, Sutskever, and Hinton in 2012 was the turning out gathering for CNNs in the PC vision group. This was the first run through a model performed so well on a truly troublesome ImageNet dataset. Using methods that are as yet utilized today, for example, information increase and dropout, this paper truly represented the advantages of CNNs and sponsored them up with record softening execution up the opposition.

2.1.2 ZF NET (2013)

[36] With AlexNet taking the show in 2012, there was a vast increment in the quantity of CNN models submitted to ILSVRC 2013. The champ of the opposition that year was a system worked by Matthew Zeiler and Rob Fergus from NYU. Named ZF Net, this model accomplished a 11.2% blunder rate. This engineering was to a greater degree an adjusting to the past AlexNet structure, yet at the same time built up some extremely keys thoughts regarding enhancing execution. Another reason this was such an awesome paper is, to the point that the creators invested a decent measure of energy clarifying a great deal of the instinct behind ConvNets and demonstrating to envision the channels and weights accurately.

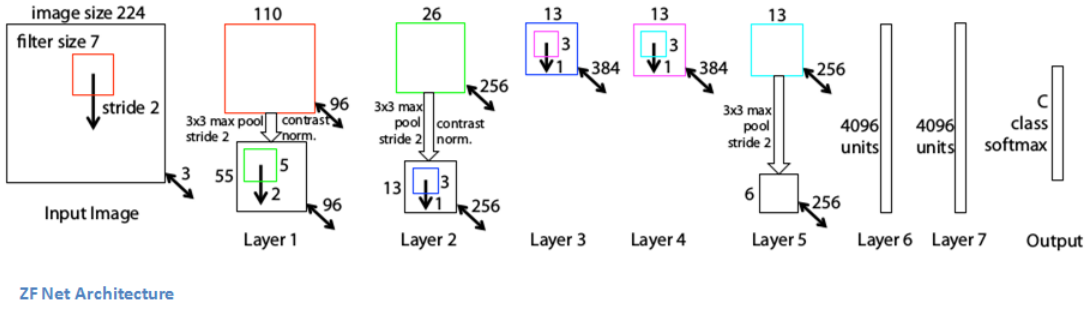


Figure 2.2 : ZF Net

In this paper titled "Picturing and Understanding Convolutional Neural Networks", Zeiler and Fergus start by talking about this reestablished enthusiasm for CNNs is

because of the openness of extensive preparing sets and expanded computational power with the use of GPUs. They additionally discuss the restricted information that analysts had on internal instruments of these models, saying that without this understanding, the "advancement of better models is lessened to experimentation". While we do as of now have a superior comprehension than 3 years prior, this still remains an issue for a great deal of scientists! The fundamental commitments of this paper are subtle elements of a somewhat changed AlexNet show and an exceptionally intriguing method for envisioning highlight maps.

Principle Points -

- Very comparable design to AlexNet, with the exception of a couple of minor adjustments.
- AlexNet prepared on 15 million pictures, while ZF Net prepared on just 1.3 million pictures.
- Instead of utilizing 11x11 estimated channels in the primary layer (which is the thing that AlexNet actualized), ZF Net utilized channels of size 7x7 and a diminished walk esteem. The thinking behind this adjustment is that a littler channel estimate in the main conv layer holds a great deal of unique pixel data in the information volume. A sifting of size 11x11 turned out to be skirting a considerable measure of important data, particularly as this is the main conv layer.
- As the system develops, we likewise observe an ascent in the quantity of channels utilized.
- Used ReLUs for their actuation capacities, cross-entropy misfortune for the mistake work, and prepared utilizing group stochastic slope plummet.
- Trained on a GTX 580 GPU for twelve days.
- Developed a representation strategy named Deconvolutional Network, which analyzes diverse component initiations and their connection to the info space. Called "deconvnet" in light of the fact that it maps highlights to pixels (the opposite a convolutional layer does).

The essential thought behind how this functions is that at each layer of the prepared CNN, you connect a "deconvnet" which has a way back to the picture pixels. An information picture is sustained into the CNN and actuations are figured at each level. This is the forward pass. Presently, suppose we need to analyze the actuations of a specific component in the fourth conv layer. We would store the enactments of this one component delineate, set the majority of alternate initiations in the layer to 0, and after that pass this element outline the contribution to the deconvnet. This deconvnet

has an indistinguishable channels from the first CNN. This info at that point experiences a progression of unpool (turn around maxpooling), correct, and channel operations for each first layer until the point that the information space is come to.

The thinking behind this entire procedure is that we need to analyze what sort of structures energize a given component delineate. We should take a gander at the perceptions of the first and second layers.

Like we talked about in Part 1, the primary layer of your ConvNet is dependably a low level element locator that will distinguish basic edges or hues in this specific case. We can see that with the second layer, we have more roundabout highlights that are being recognized. We should take a gander at layers 3, 4, and 5.

These layers demonstrate significantly more of the more elevated amount highlights, for example, puppies' appearances or blooms. One thing to note is that as you may recollect, after the primary conv layer, we regularly have a pooling layer that downsamples the picture (for instance, transforms a 32x32x3 volume into a 16x16x3 volume). The impact this has is that the second layer has a more extensive extent of what it can find in the first picture. For more data on deconvnet or the paper when all is said in done, look at Zeiler himself displaying on the subject.

ZF Net was not just the victor of the opposition in 2013, yet in addition gave incredible instinct with regards to the workings on CNNs and represented more approaches to enhance execution. The representation approach portrayed encourages not exclusively to clarify the internal workings of CNNs, yet in addition gives understanding to changes to arrange structures. The intriguing deconv representation approach and impediment tests make this one of my undisputed top choice papers.

2.1.3 VGG NET (2014)

[37] Effortlessness and profundity. That is the thing that a model made in 2014 (weren't the victors of ILSVRC 2014) best used with its 7.3% blunder rate. Karen Simonyan and Andrew Zisserman of the University of Oxford made a 19 layer CNN that entirely utilized 3x3 channels with walk and cushion of 1, alongside 2x2 maxpooling layers with walk 2. Sufficiently basic right?

Primary Points -

- The utilization of just 3x3 estimated channels is very not the same as AlexNet's 11x11 channels in the primary layer and ZF Net's 7x7 channels. The creators' thinking is that the blend of two 3x3 conv layers has a powerful responsive field of 5x5. This thusly mimics a bigger channel while keeping the advantages of littler channel sizes. One of the advantages is a lessening in the quantity of parameters. Additionally, with two conv layers, we're ready to utilize two ReLU layers rather than one.

- 3 conv layers consecutive have a successful open field of 7x7.

- As the spatial size of the info volumes at each layer diminish (aftereffect of the conv and pool layers), the profundity of the volumes increment because of the expanded number of channels as you go down the system.

- Interesting to see that the quantity of channels copies after each maxpool layer. This fortifies contracting spatial measurements, however developing profundity.

- Worked well on both picture characterization and restriction errands. The creators utilized a type of confinement as relapse (see page 10 of the paper for all points of interest).

- Built display with the Caffe tool kit.

- Used scale jittering as one information enlargement strategy amid preparing.

- Used ReLU layers after each conv layer and prepared with clump angle plummet.

- Trained on 4 Nvidia Titan Black GPUs for half a month.

VGG Net is a standout amongst the most powerful papers in my psyche since it strengthened the thought that convolutional neural systems need to have a profound system of layers all together for this various leveled portrayal of visual information to work. Keep it profound. Keep it straightforward.

2.1.5 GoogleNET(2015)

[38] You realize that thought of straightforwardness in arrange design that we just discussed? Indeed, Google sort of tossed that out the window with the presentation of the Inception module. GoogLeNet is a 22 layer CNN and was the champ of ILSVRC 2014 with a main 5 mistake rate of 6.7%. As far as anyone is concerned, this was one of the primary CNN designs that truly strayed from the general approach of just stacking conv and pooling layers over each other in a consecutive structure. The creators of the paper likewise accentuated this new model spots striking thought on memory and power use (Important note that I now and again overlook as well: Stacking these layers and including colossal quantities of channels has a computational and memory cost, and additionally an expanded possibility of overfitting).

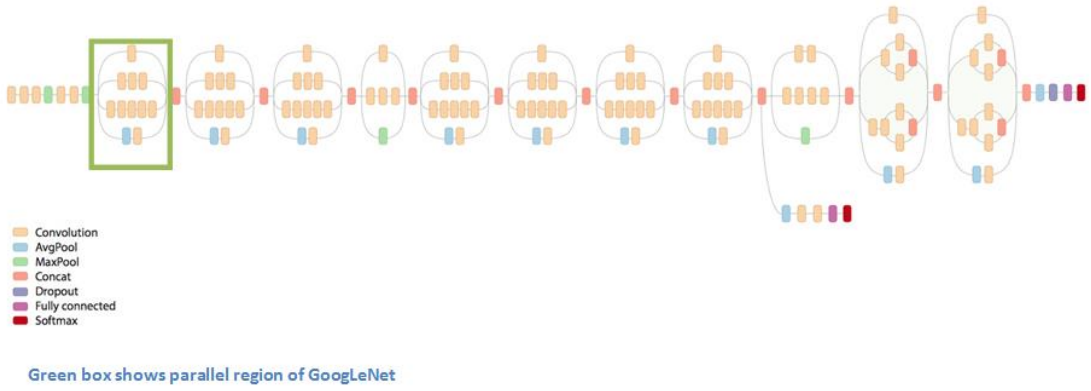


Figure 2.3 : GoogleNet

When we initially investigate the structure of GoogLeNet, we see instantly that not all things are occurring successively, as observed in past models. We have bits of the system that are going on in parallel.

This crate is called an Inception module. We should investigate what it's made of.

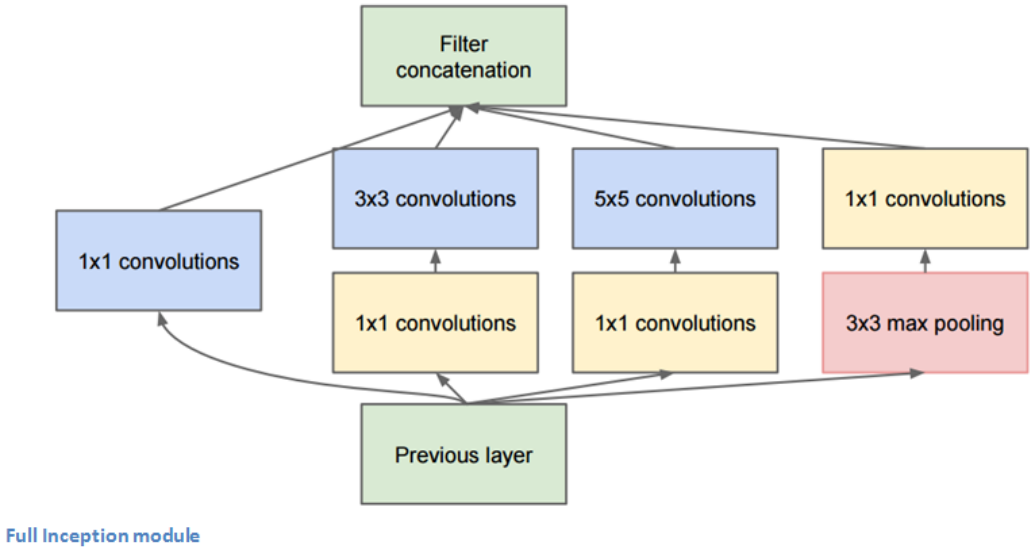


Figure 2.4 : Full Inception Module

The base green box is our info and the main one is the yield of the model (Turning this photo right 90 degrees would give you a chance to imagine the model in connection to the last picture which demonstrates the full system). Fundamentally, at each layer of a customary ConvNet, you need to settle on a decision of whether to have a pooling operation or a conv operation (there is additionally the decision of channel estimate). What an Inception module enables you to do is play out these operations in parallel. Actually, this was precisely the "credulous" thought that the creators concocted.

Presently, for what reason doesn't this work? It would prompt much excessively numerous yields. We would wind up with a to a great degree expansive profundity channel for the yield volume. The way that the creators address this is by including

1x1 conv operations before the 3x3 and 5x5 layers. The 1x1 convolutions (or system in organize layer) give a strategy for dimensionality decrease. For instance, suppose you had an information volume of 100x100x60 (This isn't really the measurements of the picture, only the contribution to any layer of the system). Applying 20 channels of 1x1 convolution would enable you to decrease the volume to 100x100x20. This implies the 3x3 and 5x5 convolutions won't have as extensive of a volume to manage. This can be thought of as a "pooling of highlights" since we are lessening the profundity of the volume, like how we diminish the measurements of stature and width with ordinary maxpooling layers. Another note is that these 1x1 conv layers are trailed by ReLU units which unquestionably can't hurt (See Aaditya Prakash's extraordinary post for more data on the adequacy of 1x1 convolutions). Look at this video for an awesome representation of the channel connection toward the end.

You might ask yourself "How does this engineering help?". All things considered, you have a module that comprises of a system in arrange layer, a medium estimated channel convolution, an expansive measured channel convolution, and a pooling operation. The system in arrange conv can extricate data about the fine grain points of interest in the volume, while the 5x5 channel can cover a vast open field of the info, and along these lines ready to separate its data also. You likewise have a pooling operation that lessens spatial sizes and battle overfitting. Over the greater part of that, you have ReLUs after each conv layer, which help enhance the nonlinearity of the system. Essentially, the system can play out the elements of these distinctive operations while as yet remaining computationally chivalrous. The paper does likewise give to a greater degree an abnormal state thinking that includes subjects like sparsity and thick associations (read Sections 3 and 4 of the paper. Still not absolutely clear to me, but rather in the event that anyone has any bits of knowledge, I'd love to hear them in the remarks!).

Principle Points -

- Used 9 Inception modules in the entire design, with more than 100 layers altogether! Now that is profound…
- No utilization of completely associated layers! They utilize a normal pool rather, to go from a 7x7x1024 volume to a 1x1x1024 volume. This spares a colossal number of parameters.
- Uses 12x less parameters than AlexNet.
- During testing, numerous yields of a similar picture were made, encouraged into the system, and the softmax probabilities were found the middle value of to give us the last arrangement.
- Utilized ideas from R-CNN (a paper we'll talk about later) for their location show.
- There are refreshed adaptations to the Inception module (Versions 6 and 7).

- Trained on "a couple of top of the line GPUs inside seven days".

GoogLeNet was one of the principal models that presented CNN layers didn't generally need to be stacked up successively. Concocting the Inception module, the creators demonstrated that an imaginative organizing of layers can prompt enhanced execution and computationally proficiency. This paper has truly set the phase for some astonishing designs that we could find in the coming years.

2.1.6 Microsoft ResNETs (2015)

[39] Envision a profound CNN engineering. Take that, twofold the quantity of layers, include a couple more, it still most likely isn't as profound as the ResNet design that Microsoft Research Asia thought of in late 2015. ResNet is another 152 layer arrange engineering that set new records in order, identification, and limitation through one unfathomable design. Beside the new record as far as number of layers, ResNet won ILSVRC 2015 with a mind boggling mistake rate of 3.6% (Depending on their aptitude and mastery, people for the most part drift around a 5-10% blunder rate. See Andrej Karpathy's incredible post on his encounters with contending with ConvNets on the ImageNet challenge).

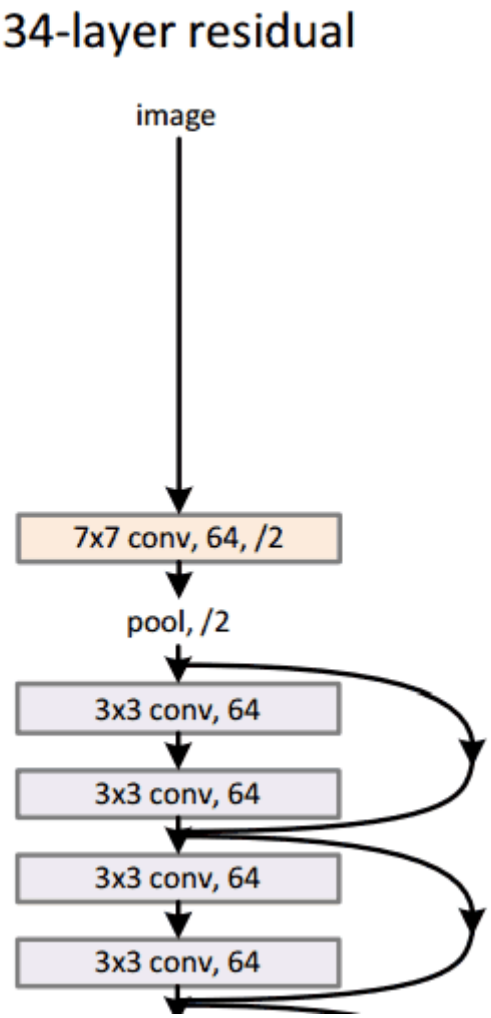


Figure 2.5 : Microsoft ResNet

The thought behind a leftover square is that you have your info x experience conv-relu-conv arrangement. This will give you some F(x). That outcome is then added to the first information x. We should call that H(x) = F(x) + x. In conventional CNNs, your H(x) would simply be equivalent to F(x) correct? In this way, rather than simply processing that change (straight from x to F(x)), we're figuring the term that you need to include, F(x), to your info, x. Fundamentally, the scaled down module appeared beneath is processing a "delta" or a slight change to the first information x to get a marginally adjusted portrayal (When we consider conventional CNNs, we go from x to F(x) which is a totally new portrayal that doesn't keep any data about the first x). The creators trust that "it is less demanding to upgrade the leftover mapping than to advance the first, unreferenced mapping".

Another explanation behind why this leftover square may be viable is that amid the regressive go of backpropagation, the slope will stream effortlessly through the chart since we have expansion operations, which circulates the angle.

Primary Points -

- "Ultra-profound" – Yann LeCun.
- 152 layers
- Interesting note that after just the initial 2 layers, the spatial size gets compacted from an info volume of 224x224 to a 56x56 volume.
- Authors guarantee that a guileless increment of layers in plain nets result in higher preparing and test blunder (Figure 1 in the paper).
- The amass attempted a 1202-layer organize, however got a lower test precision, apparently because of overfitting.
- Trained on a 8 GPU machine for a little while.

3.6% mistake rate. That itself ought to be sufficient to persuade you. The ResNet display is the best CNN engineering that we right now have and is an incredible development for the possibility of remaining learning. With mistake rates dropping each year since 2012, I'm suspicious about regardless of whether they will go down for ILSVRC 2016. I accept we've come to the heart of the matter where stacking more layers over each other wouldn't bring about a generous execution support. There would need to be innovative new structures like we've seen the most recent 2 years.

2.1.7 Region Based CNNs (R-CNN) (2013) , Fast R-CNNs & Faster RCNNs(2015)

[40,41] Some may contend that the approach of R-CNNs has been more impactful that any of the past papers on new system models. With the main R-CNN paper being refered to more than 1600 times, Ross Girshick and his gathering at UC Berkeley made a standout amongst the most impactful progressions in PC vision. As clear by their titles, Fast R-CNN and Faster R-CNN attempted to improve the model quicker and suited for present day question recognition errands.

The reason for R-CNNs is to take care of the issue of question recognition. Given a specific picture, we need to have the capacity to draw jumping boxes over the greater part of the articles. The procedure can be part into two general segments, the area proposition step and the arrangement step.

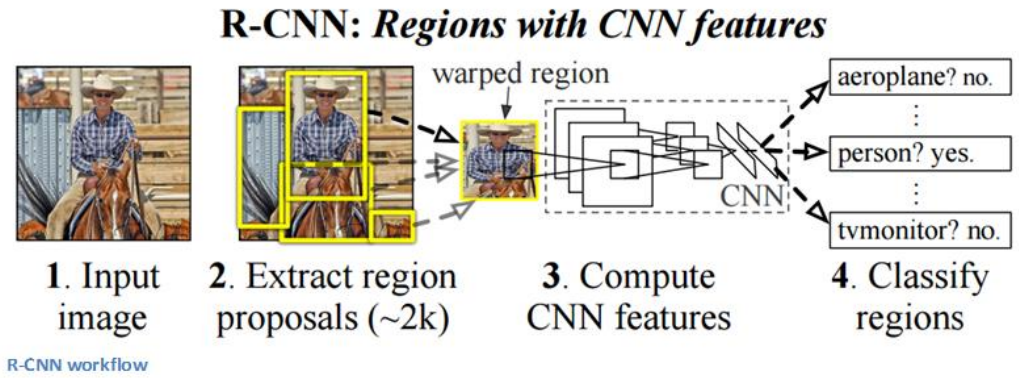


Figure 2.6 : R-CNN

The creators take note of that any class skeptic area proposition technique should fit. Specific Search is utilized as a part of specific for RCNN. Specific Search plays out the capacity of creating 2000 unique districts that have the most astounding likelihood of containing a protest. After we've thought of an arrangement of locale recommendations, these proposition are then "twisted" into a picture estimate that can be encouraged into a prepared CNN (AlexNet for this situation) that concentrates a component vector for every area. This vector is then utilized as the contribution to an arrangement of direct SVMs that are prepared for each class and yield an order. The vector likewise gets encouraged into a jumping box regressor to get the most exact directions.

Non-maxima concealment is then used to smother bouncing boxes that have a critical cover with each other.

Fast R-CNN

Upgrades were made to the first model in light of 3 primary issues. Preparing took different stages (ConvNets to SVMs to jumping box regressors), was computationally costly, and was greatly moderate (RCNN took 53 seconds for every picture). Quick R-CNN could take care of the issue of speed by essentially sharing calculation of the conv layers between various proposition and swapping the request of producing district recommendations and running the CNN. In this model, the picture is first sustained through a ConvNet, highlights of the locale recommendations are gotten

from the last component guide of the ConvNet (check segment 2.1 of the paper for more points of interest), and ultimately we have our completely associated layers and also our relapse and order heads.

Faster R-CNN

Faster R-CNN attempts to battle the fairly complex preparing pipeline that both R-CNN and Fast R-CNN showed. The creators embed a district proposition organize (RPN) after the last convolutional layer. This system can simply take a gander at the last convolutional highlight guide and deliver area proposition from that. From that

stage, an indistinguishable pipeline from R-CNN is utilized (ROI pooling, FC, and afterward grouping and relapse heads).

Having the capacity to confirm that a particular question is in a picture is a certain something, however having the capacity to verify that protest's correct area is a tremendous hop in information for the PC. Faster R-CNN has turned into the standard for protest recognition programs today.

2.2 Feature Extraction

2.2.1 ASM: Active shape Model

[5] (ASM) proposed by Cootes et al. is a component coordinating technique in light of a measurable model. An ASM is contained a Point Distribution Model (PDM) taking in the progressions of substantial shapes, and various flexible models catching the dim levels around various milestone include focuses.

[6] Figure 16 demonstrates a case with the ASM include extraction strategy in , defined by 58 facial historic point highlight focuses. The ASM strategy incorporates two stages. In the first place, shape models are worked from the preparation tests with some commented on milestone include focuses. At that point, nearby surface models for every milestone include point are likewise fabricated. Second, as per the two building models, an iterative inquiry technique to twist the model illustration can be finished.

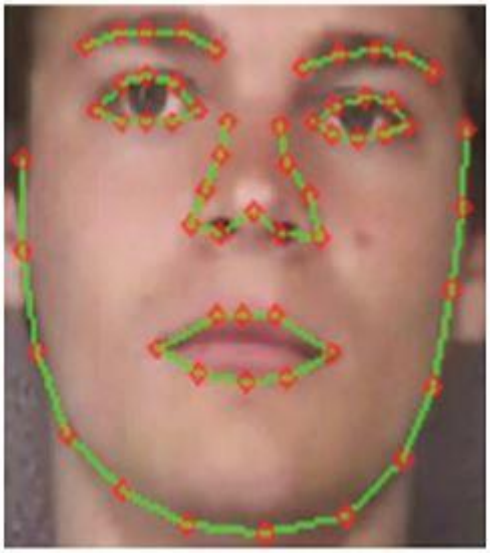


Figure 2.7 : Feature Extraction from Face

[7] Shbib and Zhou utilized the geometric dislodging among the anticipated ASM highlight point arranges and the mean state of ASM as facial highlights for FER. [8] As of late, Anderson et al. introduced an improved adaptation of ASM called dynamic

shape and factual models (ASSM) for confront acknowledgment, which has potential applications for FER.

2.2.2 AAM: Active Appearance Models

[9] (AAM) was created by Cootes et al. in 2001. AAM basically expands ASM by catching the shape and surface data together. In detail, AAM first assembles a factual model in view of preparing information for measurable investigation, and afterward utilize this factual model to execute fitting computation for testing information. Distinctive with ASM, AAM not just exploits the worldwide shape and surface data, yet in addition conducts factual investigation on neighborhood surface data in order to find out the connections amongst shape and surface data.

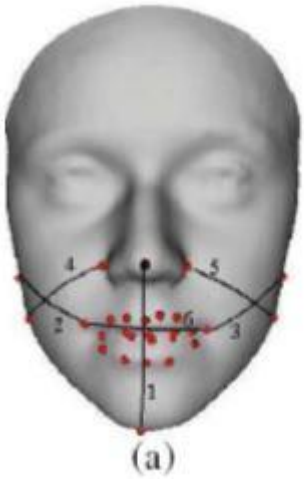
[10] Cheon and Kim exhibited a FER technique by utilizing differential-AAM and complex learning. To begin with, the distinction of AAM parameters between the info pictures and the reference pictures, (for example, unbiased articulation pictures) is ascertained to remove the differentialAAM highlights (DAFs). Second, complex learning techniques are utilized to implant the DAFs on the smooth and nonstop component space. At long last, the info outward appearance is identified.

As of late, a few propelled renditions of AAM have been additionally grown, for example, histogram of arranged angle (HOG)- based AAM [11], thick based AAM [12], relapse based AAM [13]. It is an intriguing undertaking to research the execution of these as of late created AAM variations on FER.

2.2.3 SIFT: Scale-invariant Feature Transform

[14,15]. (SIFT) is a nearby picture descriptor for picture based coordinating proposed by David Lowe .The SIFT highlights are invariant to picture scaling, interpretation, and pivot, and in part invariant to enlightenment changes and affine or threedimensional (3D) projection.

[16] Figure 17 gives a case of the SIFT highlight extraction technique utilized as a part of Berretti et al ,in which they took facial historic points situated in vital morphological districts of the face as key focuses, and after that the SIFT include extractor was executed on these found key indicates all together get the SIFT descriptor.



(a)

component	#points
nose	2
mouth	20
face contour	5
line 1	12
line 2	16
line 3	16
line 4	16
line 5	16
line 6	9

(b)

Figure 2.8 : SIFT Features

[17] As of late, Soyel and Demirel gave a separation of scale-invariant component change (D-SIFT) technique, which can adequately settle on choices on the general appearance highlights.

[18] Li et al.introduced another scale-invariant element change called GA-SIFT for multispectral picture utilizing geometric variable based math (GA). At first, in view of the hypothesis of the GA, a novel portrayal of multispectral pictures with ghastly and spatial data was exhibited. Second, finding the scale space of a multispectral picture was given. Third, like SIFT, GA-based contrast of Gaussian pictures were gotten. At last, the element focuses can be recognized and depicted in view of the hypothesis of GA.

2.2.4 Appearance-based techniques

Appearance-based strategies mean to utilize the entire face or specific areas in a face picture to reflect the hidden data in a face picture, particularly the inconspicuous changes of the face, for example, wrinkles and wrinkles.

2.2.4.1 LBP: Local Binary Pattern

[19] Up until now, there are principally two agent appearance-based component extraction techniques, i.e. neighborhood twofold examples (LBP) and Gabor wavelet portrayal (LBP) is a compelling surface depiction administrator, which can be utilized to gauge and concentrate the contiguous surface data in a picture. The benefit of utilizing the LBP administrator is that the LBP administrator has a decent revolution invariance and dim invariance, and beats the issues of disequilibrium relocation, pivot and enlightenment in a picture. Besides, the LBP administrator has a generally straightforward computation. Figure 18 demonstrates a case of the LBP highlight extraction for FER .

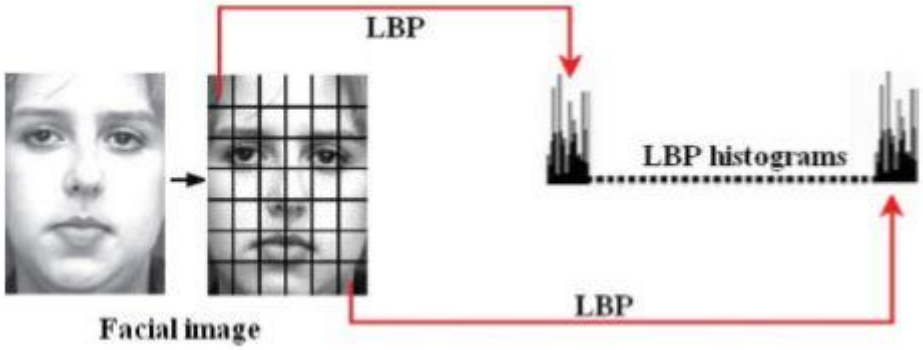


Figure 2.9 : LBP - Local Binary Pattern

[20] The utilized LBP include extraction technique ,contains three critical advances. At first, a facial picture was isolated into different non-covering pieces. Second, LBP histograms were worked out for each square. Third, the square LBP histograms were connected into a solitary vector spoke to by the LBP code.

[21,22] In our past works we researched the execution of the LBP administrator with dimensionality lessening strategies, for example, neighborhood fisher discriminant investigation on FER undertakings.

[23]. As of late, a few variations of the LBP administrator can be found in the writing .Till now the ordinary LBP variations contain volume nearby paired examples (VLBP) [24], LBP on three orthogonal planes (LBP-TOP) [24], nearby directional examples (LDP) [25], neighborhood transitional examples (LTP) [26], et cetera.

[27] As of late, Li et al. has proposed the polytypic multi-square nearby paired examples (P-MLBP) for completely programmed 3D FER. The P-MLBP includes both the component based sporadic divisions to precisely speak to the outward appearance, and incorporate the profundity and surface data of 3D models to improve facial highlights.

[28] Gabor wavelet portrayal is a traditional technique to extricate outward appearance highlights. In detail, a picture is filtered by an arrangement of filters, and the filtered results can reflect the relationship (slope, surface connection, and so forth.) between neighborhood pixels. Gabor wavelet portrayal strategy has been generally utilized for outward appearance include extraction. It can recognize multi-scale, multidirection changes of surface, and littly affects light changes. Figure 19 displays a case of Gabor wavelet portrayal utilized as a part of , in which the aggregate 18 Gabor bits at three scales and six introductions were utilized. Liu et al.

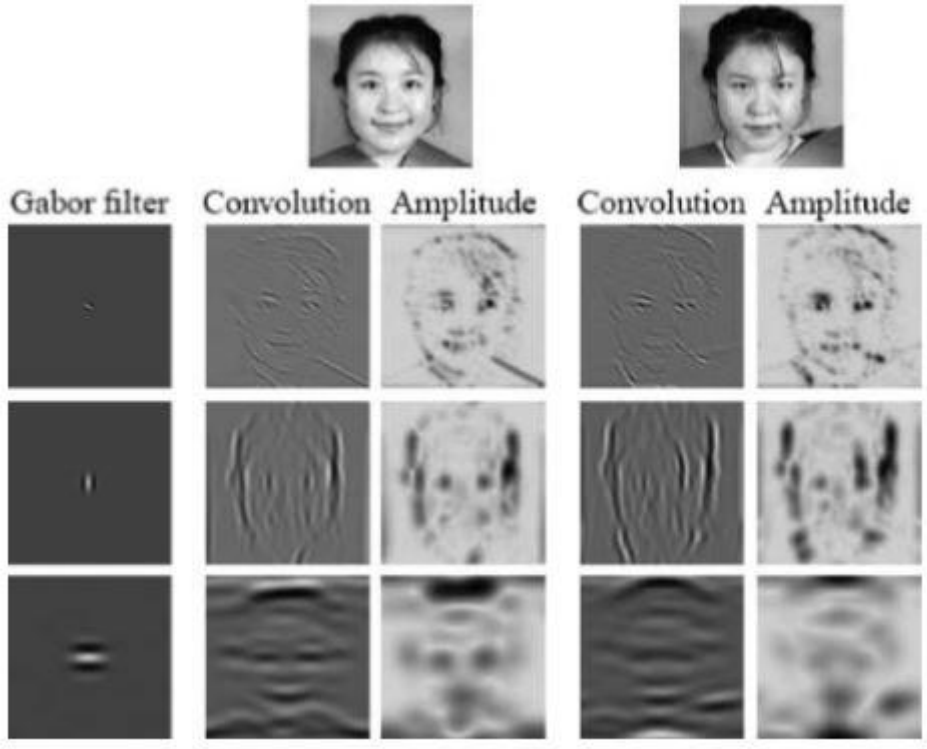


Figure 2.10 : Gabor Wavelet Potrayal

[29] proposed a FER technique in light of Gabor wavelet highlights and piece primary part examination (KPCA). In this plan, they utilized a neighborhood Gabor filter to supplant the customary Gabor filter, bringing about the reality it can accelerate the calculation speed.

[30] Gu et al. performed FER by utilizing the outspread encoding of neighborhood Gabor highlights and classifier blend. In this investigation, the information pictures were first subjected to nearby, multi-scale Gabor-filter operations, and afterward the subsequent Gabor deteriorations were utilized to be encoded with outspread frameworks.

[31] As of late, Owusu et al. introduced a neural-AdaBoost-based FER framework in which Gabor include extraction strategies were utilized to separate a substantial number of facial highlights speaking to different facial twisting examples.

Dynamic picture groupings reflect the constant procedure of outward appearance developments. Outward appearance highlights for dynamic picture arrangements are primarily spoken to by misshapening and facial muscle developments. At present, two prominent element extraction strategies for dynamic picture arrangements are given as takes after: optical flow, and highlight point following.

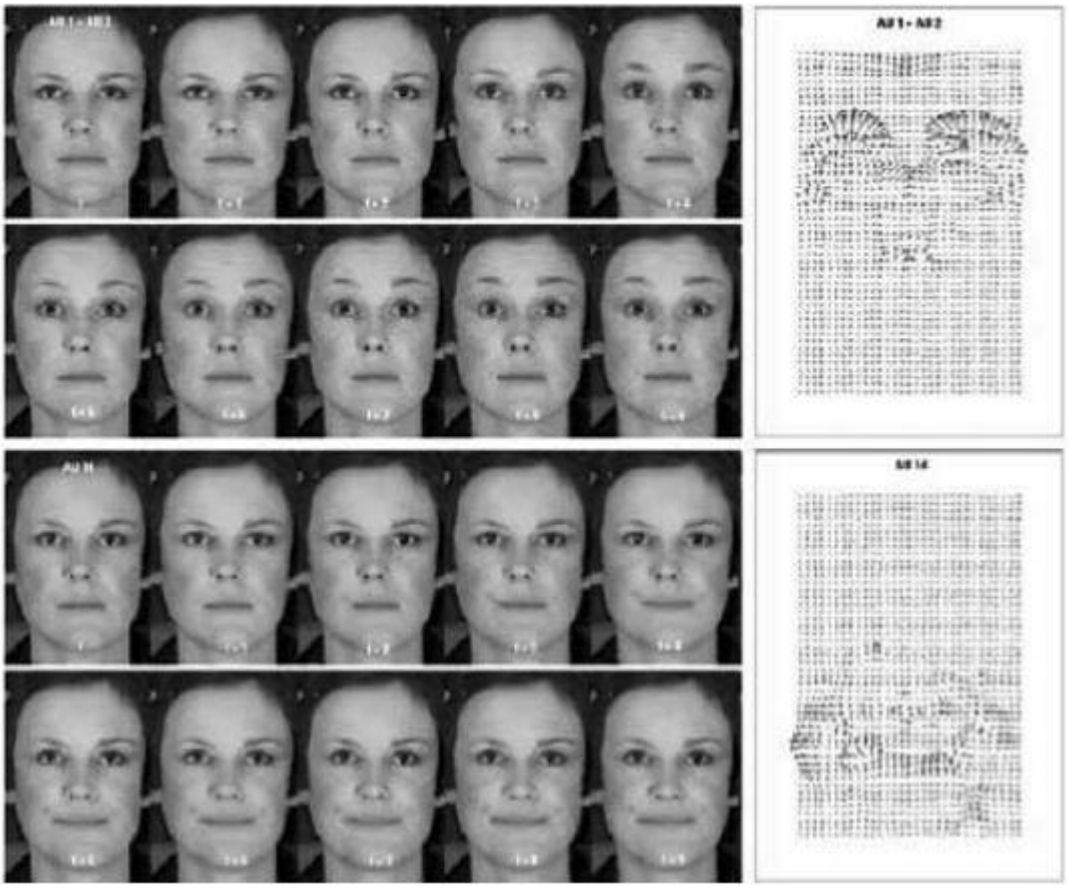


Figure 2.11 : Optical Flow Extraction Strategy

[32] Optical flow Negahdaripour redefines the optical flow technique as geometry and radiation changes of dynamic pictures. The essential rule of the optical flow technique is that every pixel in a picture is allocated to a speed vector. These speed vectors frame a movement field for a picture. In a movement minute, the picture point relates to the real protest point. The optical flow technique has an undeniable favorable position, that is, the optical flow not just conveys the movement data of the objective, yet additionally has the rich data about the 3D structure of the objective. In the field of FER, the optical flow strategy is broadly used to extricate outward appearance highlights from dynamic picture arrangements since it features facial misshapening and reflects the movement pattern of picture successions. Figure 20 demonstrates a case of the optical flow include extraction strategy utilized as a part of , which was performed on two outward appearance arrangements.

[33] Lien broke down all encompassing face movement by methods for wavelet-based multi-determination thick optical flow, and afterward figured out PCA-basedeigenflows both in level and vertical bearings for a compacter portrayal of the subsequent flow fields.

[34] Yacoob et al. utilized the optical flow field and the angle field between progressive edges to speak to the fleeting and spatial varieties of pictures. At that

point, as per the progressions of the movement vectors of highlights, the facial muscle developments were figured to characterize diverse appearance.

[35] Sanchez et al.looked at deliberately two optical flow-based strategies for FER. One was featural and expected to choose a lessened arrangement of exceptionally discriminant facial focuses. The other was comprehensive and utilized considerably more focuses that were consistently conveyed on the focal face area.The element point following techniques regularly select some element focuses with vast changes toward the edges of eyes and mouth. At that point, following these focuses will have the capacity to get facial element relocation or disfigurement data.

.[36] Figure 21 introduces a case of the element point following strategy utilized as a part of Pantic et al ,in which 15 highlight focuses were chosen in light of facial activity coding System (FACS), and afterward the molecule filter was utilized to track the developments of highlight focuses in picture arrangements.

Figure 2.12 : Facial Activity Coding System - FACS

[37] Tie et al. proposed a strategy which could consequently separate 26 reference focuses in a facial model from video groupings, and followed the reference focuses through various molecule filters.

[38] Tooth et al. utilized the remarkable facial guide following technique toward separate notable data from video groupings however did not depend on any subjective preprocessing or extra client provided data to choose outlines with top appearance.

2.3 Facial Expression Classification – Drawing Comparisons

To check the execution of various classifiers on FER, we performed FER investigates two well known databases, i.e. the JAFFE database [32] and the Cohn-Kanade database [33]. These two databases contains seven outward appearance, i.e. outrage, happiness, bitterness, impartial, amazement, nauseate and fear.

The JAFFE database has 213 pictures of female outward appearance. Each picture has a determination of 256 x 256 pixels. Some example pictures are appeared in Figure 8. The CohnKanade database contains 100 college understudies. Each picture has a determination of 640x490 pixels. Figure 9 introduces some specimen pictures from the CohnKanade database. As done in [21,34], on the CohnKanade database we chose 320 picture arrangements from 96 subjects, with 1 to 6 feelings for every subject. For each succession, the unbiased face and one pinnacle outlines were utilized for prototypic articulation acknowledgment, giving altogether 470 pictures (32 outrage, 100 bliss, 55 trouble, 75 astound, 47 fear, 45 disturb and 116 impartial).

Methods	Accuracy (%)
HMM	78.64
Naive-Bayes	70.57
ANN	68.09
KNN	80.95
SVM	79.88
SRC	84.76

Table 2.1 : Performance Comparison of Different Classification methods with the LBP features on the JAFFE Database

For facial component portrayal, the LBP [19] highlights were separated because of its calculation effortlessness and promising execution. As indicated by the standardized estimation of the eye separate, a resized picture of 110 x 150 pixels was edited from unique pictures. Like the settings in [21,34], we utilized the 59-container operatorLBPu2 P;R, and isolated the edited pictures of 110 x 150 pixels into 18x21 pixels districts, yielding an element vector length of 2478 (59 x 42) spoke to by the LBP histograms. A 10-overlay cross-approval plot is executed in seven-class FER tests, and the normal acknowledgment comes about are accounted for.

Methods	Accuracy (%)
HMM	94.76
Naive-Bayes	93.81
ANN	93.45
KNN	96.22
SVM	95.24
SRC	97.14

Table 2.2 : Performance Comparison of Different Classification methods with the LBP features on the Cohn-Kanade Database

For HMM, we utilized a seven-state discrete HMM show with the leftright structure, in which each state compared to one outward appearance. For ANN, RBFNN with a three-layer feedforward arrange containing one information layer, one shrouded layer and also one yield layer, is utilized for its computational straightforwardness. The quantity of concealed layer neurons in RBFNN is set to be the quantity of preparing tests. The objective of preparing mistake is 0.0001. For the BN, we utilized the innocent Bayes classifier. For KNN, we set K to be 1 for its fantastic execution. For SVM, we utilized the LIBSVM bundle, accessible at http://www.csie.ntu.edu.tw/»cjlin/libsvm, to play out the SVM calculation with the direct part work, one-against-one for multi-class issues. The examination stage is Intel CPU 2.10 GHz, 1G RAM memory, MATLAB 2012a.

	Anger (%)	Joy (%)	Sadness (%)	Surprise (%)	Disgust (%)	Fear (%)	Neutral (%)
Anger	93.33	0	6.67	0	0	0	0
Joy	0	100	0	0	0	0	0
Sad	3.22	3.22	74.19	3.22	3.22	6.48	6.45
Surprise	0	3.45	3.45	82.76	0	10.34	0
Disgust	10.35	0	6.89	0	82.76	0	0
Fear	0	0	12.52	3.12	9.37	71.87	3.12
Neutral	3.45	0	0	6.89	0	0	89.66

Table 2.3 : Confusion Matrix of recognition results of SRC with LBP features on the JAFFE Database

Tables 1 and 2 independently exhibit the acknowledgment aftereffects of six diverse classification strategies, including HMM, guileless Bayes, ANN, KNN, SVM, SRC on the JAFFE database and the Cohn-Kanade database, As appeared in

Tables 1 and 2, it can be seen that SRC gets the best execution on FER assignments, trailed by KNN, SVM, HMM, guileless Bayes, ANN. This confirms the legitimacy

and superior of SRC for FER. To additionally demonstrate the point by point exactness of every articulation, Tables 3 and 4 demonstrate the disarray lattice of acknowledgment consequences of SRC on the JAFFE database and the Cohn Kanade database, separately.

	Anger (%)	Joy (%)	Sadness (%)	Surprise (%)	Disgust (%)	Fear (%)	Neutral (%)
Anger	90	0	0	0	0	0	10
Joy	0	100	0	0	0	0	0
Sad	3.33	0	90	0	0	0	6.67
Surprise	0	0	0	100	0	0	0
Disgust	0	0	0	0	100	0	0
Fear	0	0	0	0	0	100	0
Neutral	0	0	0	0	0	0	100

Table 2.4 : Confusion Matrix of recognition results of SRC with LBP features on the Cohn-Kanade Database

CHAPTER 3 : PRESENT WORK

3.1 PROBLEM FORMULATION

Facial Expression is an imperative method of communicating and deciphering passionate states and mental conditions of individuals. In early brain research, Mehrabian [1] has discovered that lone 7% of the entire data human communicates is passed on through dialect, 38% through discourse, and 55% through outward appearance. Subsequently, through outward appearance a lot of profitable data can be acquired in order to distinguish people's cognizance and mental exercises. Outward Facial Expression Recognition (FER) plans to build up a programmed, efficient, precise framework to recognize outward appearance of people with the goal that human feelings can be comprehended through outward appearance, for example, satisfaction, misery, outrage, fear, shock, appall, and so forth. Amid the most recent two decades, programmed FER has pulled in developing considerations in numerous fields, for example, Computer vision, design acknowledgment, and artificial knowledge, inferable from its potential applications to common human computer collaboration (HCC), human feeling investigation, intuitive video, picture ordering and recovery, and so forth.

For the most part, an essential FER framework will be developed with two noteworthy advances: using Facial feature extraction and Facial Expression classification on:

1. Mixed Emotion Recognition
2. Fake Expression Detection

3.2 OBJECTIVES OF STUDY

Thus, this dissertation just concentrates on giving late advances on these two stages, i.e. facial feature extraction, and Facial Expression classification on FER errands. Some past work [2,3,4] on checking on FER exist in the most recent decades, which is depicted as a current progress, particularly from 2013 to 2017, on FER.

Objectives of this Study will include :

1. We will perform FER investigation to show a relative examination of various classification techniques – a deep comparative study.
2. In all the above papers the methodologies proposed focused only on 7 basic Emotions ,that are : Happy, Sad ,Angry , Disgust, Neutral ,Fear & Surprise . So we will include some more features with mixed emotions detection like – “Happily Surprised” & “Angrily Disgusted”.

3.3 RESEARCH METHEDOLOGY

These are steps that we're going to be going through if these images are optimal. Step one is Formation of Convolutional Layer . Step number two is max pooling . Step number three is flattening and step number four is a full connection them in great detail and exactly what they're doing. Lucen's original paper that gave rise to an emotional neural networks. It's called gradient based learning applied to documentary cognition.

Figure 3.1 : Methodology Steps

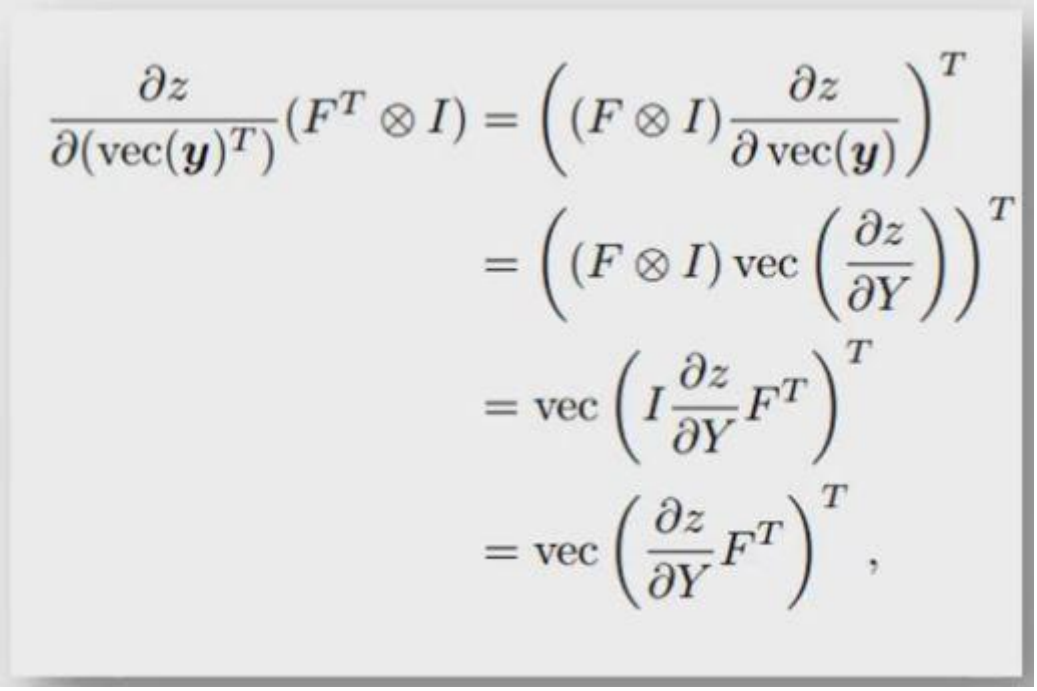
3.3.1 Step 1 : Convolution

So this is the convolution function and we try to stay away from mathematics and keep things
intuitive.

$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau$$

Figure 3.2 : Mathematical Formula for Convolution

But I couldn't help but share this formula for we because it is so simple a convolution is basically a combined integration of the two functions and it shows we how one function modifies the other or modifies the shape of the other and if we've done any signal processing or electrical engineering or a profession where signal processing is required we would have inevitably come across a conclusion function. It is quite popular now. Once again we're going to keep the mathematics lights or keep them separate. And if we'd like to get into the math behind the convolutional neural networks a great additional read is Introduction to convolutional neural networks by Jianxin Wu who is a professor at Nanjing University in China.



$$\begin{aligned}
\frac{\partial z}{\partial(\operatorname{vec}(\boldsymbol{y})^T)}(F^T \otimes I) &= \left((F \otimes I)\frac{\partial z}{\partial \operatorname{vec}(\boldsymbol{y})}\right)^T \\
&= \left((F \otimes I)\operatorname{vec}\left(\frac{\partial z}{\partial Y}\right)\right)^T \\
&= \operatorname{vec}\left(I\frac{\partial z}{\partial Y}F^T\right)^T \\
&= \operatorname{vec}\left(\frac{\partial z}{\partial Y}F^T\right)^T,
\end{aligned}$$

Figure 3.3 : Mathematical Explanation

This paper was published literally days ago like five or six days ago and it is oriented specifically at people who are starting out at beginners who are getting to know convolutional neural networks so the mathematics there should be accessible actually e-mailed Professor Johnson. His whole goal is to make or break the complex things down so that people who are new to this field can understand.

So what is a good solution in intuitive terms here on the left. We've got an input image as we discussed that's how we're going to look at images just ones and zeros to simplify things. And we can see the smiley face there. Then we've got a feature detector so feature detectors a three by three Matrix. Does it have to be three by three. No it doesn't. Alex net I think uses seven by seven. And then some other one of those other famous ones uses like five by five feature detectors.

They can be different but usually we'll see that they are three by three and they are we know reasons to make them three by three so we're going to stick to the conventional

way. Having a three by three feature detector also the feature detectors called these are important terms because we might come across them.

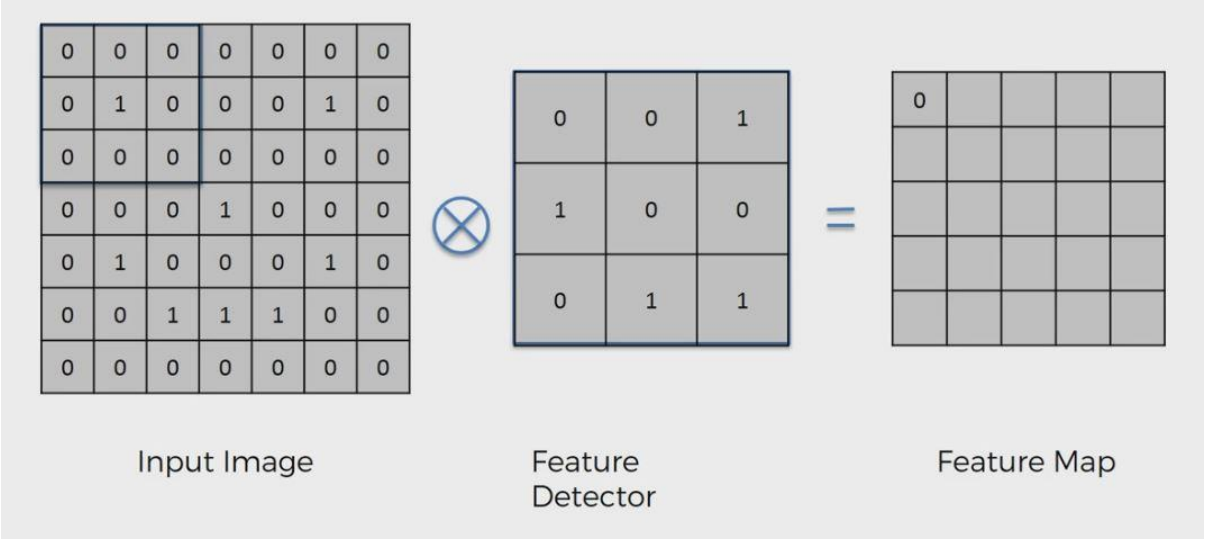


Figure 3.4 : Feature Detector - Extracting Features

There are many different terms for the feature detector but the most common ones are feature detector . Or we might hear it being called kernel or we might hear it being called Filter.

So we're going to be using either filter or a feature detector interchangeably but just bear it has those names and a coalition operation is signified by an X in a circle. Just as we saw in the formulas before and here what happens is on an intuitive level or just think of it in terms of what is actually happening in the background rather than the mathematics.

Well we take this feature detector or filter and we put it on your image like we see on the left. So we cover the for instance in this case the top left corner the nine pixels in the top left corner and we basically multiply each a valuable value so respect to value so the top 0 by the top left value by the top left value then basically is in position of a 1 one by position about 1 1 position number or 0 1 0 1 0 2 by 0 2 and so on. So it's element wise multiplication on these matrices. And then we add up the result. So in this case nothing matches up so it's always either 0 by 0 0 or by 1. So the result is zero. And here we can see that one of them matched up one on the left matched up. And therefore we've got a 1 here.

Nothing matched up nothing matched up nothing matched up.Then we move on to the next throw so and step at which we're moving this whole filter is called the stride. So here we have a stride of one pixel.

Here we can see again something matched up the bottom right corner matched up against stride but a bottom one in the middle matched up here top right hand match up

the nothing measure. The stride is one. You can change the stride. You can make it one two. You're going to get three whatever we like. The Eventually the one that works well is usually or two. So that's what people stick to. And we'll talk about what the stride is . So here we've got so we're matching absolutely when I hear we can see we've got two because two of them matched up and so on and so on. So on there we go there's another one that matched up there we go and we're done. So that's what have we created.

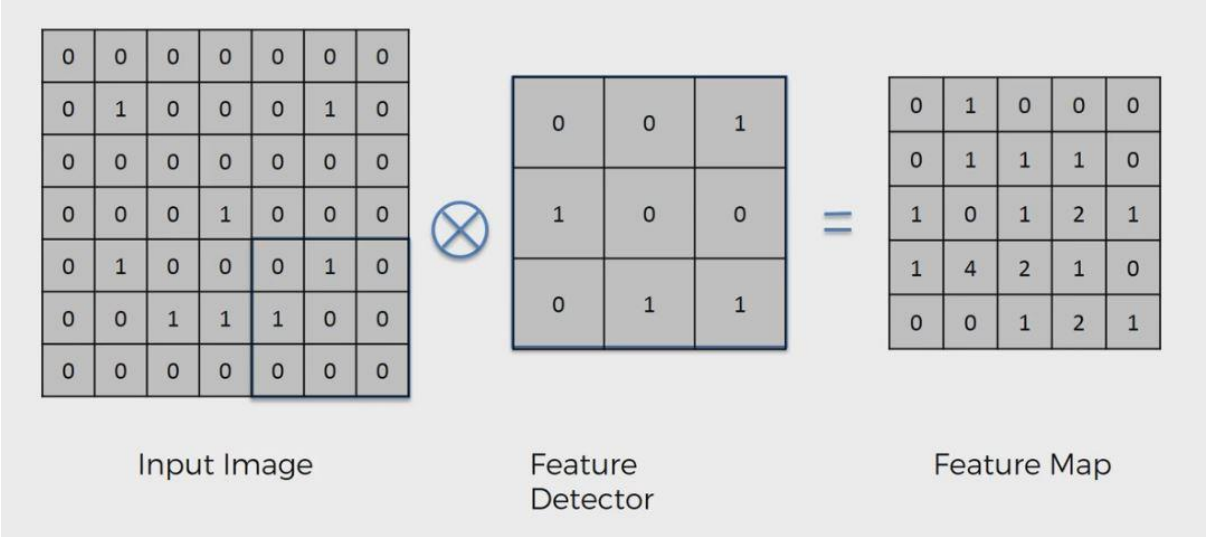


Figure 3.5 : Feature Map

The image on the right is called a feature map also has several terms it also can be called sometimes it can Vold feature.

So in your convolution operation operator to something it doesn't become convoluted it becomes convolved and it has sometimes like I think to myself in the wrong way but it is the correct term is convolved is a kind of old feature or it can also be called the activation map.

But we're going to be calling it a feature map in this course so it can be called any one of those things and what have we done here. Well as we can see we've reduced the size of the image. That's number one and that's the important thing I wanted to mention about your input image and the feature text and the stride. If we have a stride of one we can see the image reduced a bit but if we have a right to the image is going to produce more so the feature is going to be even smaller. And that's a very important function of the feature detector of this whole convolution step is to make the image smaller because that'll be it'll be easier to process it and it'll be just faster. It will and we'll be just foster because imagine like here we've got a what a seven by seven image but imagine if we have a proper photo right. Or if we have like a 256th on 56 pixel image that's it's a huge number of pixels I CHONE if it is x squared or like let's say we have a 300 but 300 pixels. So we don't get confused with the R.G. B 256 has to say like we have a 300 by 300 image in terms of size and pixels. Then we have 300

square number of pixels that's a huge number and therefore feature detectors will reduce the size of the image and therefore stride of two is actually beneficial.

But then the question is do we lose information. Are we losing information when we're applying the feature detector. Well some information we are losing of course because we have less values and of resulting matrix. But at the same time the purpose of the feature detector is to detect certain features certain parts of the image that are integral. And so for instance if we think about it this way like the feature detector has a certain pattern on it.

The highest number in your feature map is when that pattern matches up.In fact the highest number we can get is in an all simplified example is when the feature is that it matches exactly and we can see that number four we have in our feature map that's exactly. So if we look at it here that's exactly where this feature detector because there's only four ones and it matched perfectly so we can see this this part over here.

So the feature was detected here. And as we discussed at the very start of this section that it features is how we see things is how we recognize it. We don't look at every single pixel so to speak in what we see on an image or in real life.

We don't look at every single picture we look at features we look at the nose the hats the the feather the eyes under the little black marks under the cheetah's eyes to distinguish between a cheetah and a leopard or the shape of the train.We don't to distinguish between a bullet train and normal train and so on so we don't look at everything we look at features and that's what we're preserving and that's what the feature map helps us preserve.

Actually that's what it allows us to bring forward and get rid of all of the unnecessary things that even as humans we don't process so much information going into your eyes that at any given time like gigabytes of information if we look at every single dot if not terabytes of information going into your eyes per second and still we're able to proceed because we get rid of what is unnecessary only focus on the important features features that are important to us and that is exactly what the feature does.

So now moving on this is our input image and we we create a feature map so the front one let's say the front one is the one we just created but then how come there's many of them. But we create multiple feature maps because we use different filters. And that's another way that we preserve lots of the information so we don't just have one feature map we look for certain features and then or basically the network decides through its training and this is something we'll discuss towards the end of the section through his training it decides which features are important for certain types or certain categories and it looks for them and therefore will have different filters and we'll talk about filters just now. But basically Ill apply these filters so to get this feature map it applied a filter like the one we saw but then to get this feature Mabbett apply a different filter to get this feature up apply a different filter and so on. And so basically it just creates these feature maps.

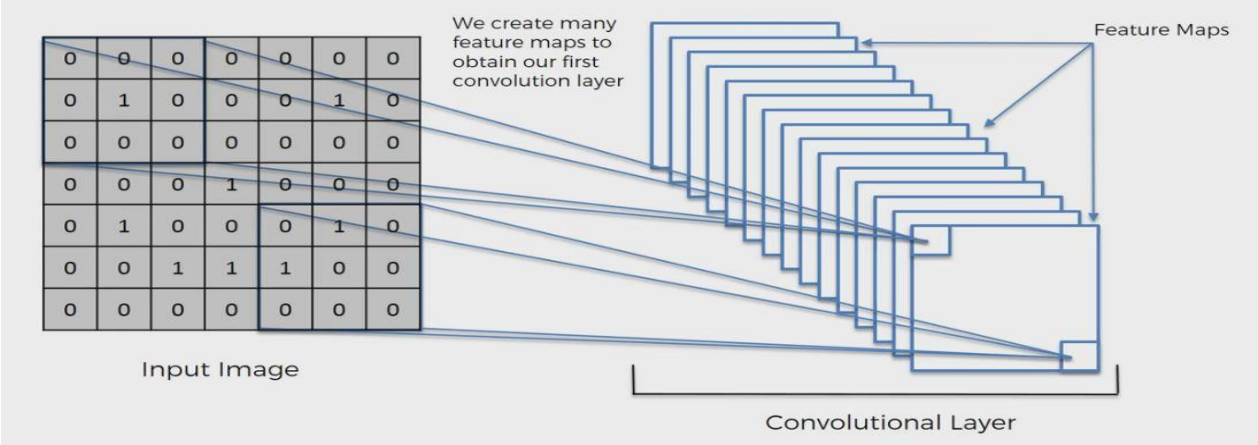


Figure 3.6 : Convolotional Layer

And actually that's why personally I think the term feature detector is better than filters. Remember we're here we have this filter which we also can call a feature detector Well actually the word feature detector I think is better suited. And the reason for that is that's what the purpose is right. We don't want to just we don't want to just filter out our image. But even though that's a whole that's the same same just a question of terminology.

But basically we want to detect features. In this in this layer we're going to our own this feature map we've detected where certain features are in the image and this feature map we've detected where certain other features are where a certain specific feature is located and this feature map will be detected where a certain other feature is located on the image.

But basically they have some valuable examples in their documentation and here they have a picture of the Taj Mahal and we can choose which filter we want to apply. So if we download this program and we upload a photo into it and then we can actually start a conversion matrix and apply filters and we will see that these things these English matrices actually applied in image processing and design and so on. So let's have a look at what we get what we get so.

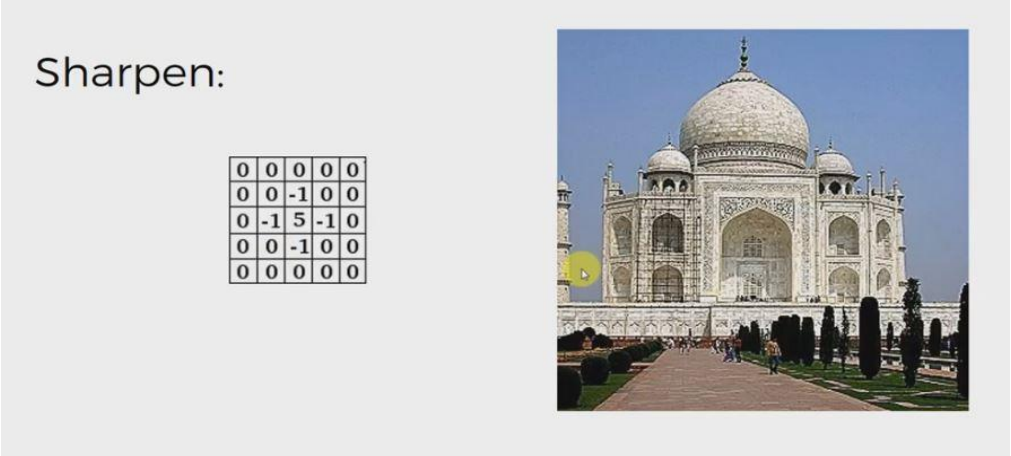


Figure 3.7 : Sharpen Feature

So if we apply this filter five in the middle minus one one is one one is one minus one. You can see that it sharpens the image. And so this is it's quite intuitive if we think of it. So 5 is the pixel of the main pixel like in the middle of the of the filter or the feature detector and then minus one minus one one just one just kind of like reduces the pixels around the a in an intuitive sense.

Then blur, so basically takes equal significant gives equal significance to all of the pixels are all the one in the center and therefore it combines them together and we get a blur edge enhance. So here we can see that's minus one and one and then we get zeros right. So we did delete to remove the pixels around the main one in the middle and we only keep this one at a minus one and it gives we an edge and this was a bit harder to understand how it works.

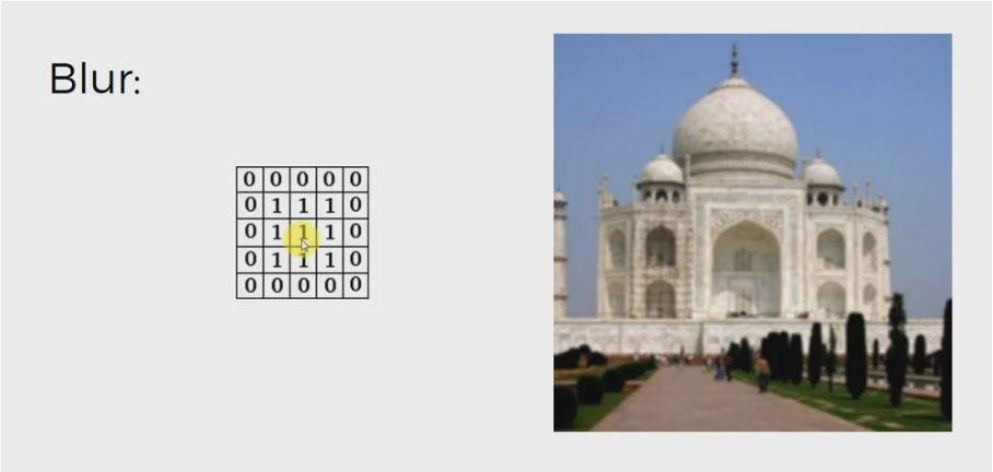


Figure 3.8 : Blur Feature

Like probably harder just to think of it intuitively edge detect. So this one probably makes more sense. So we take them a middle one. You reduce the middle one. The Probably like the strength of the middle pixel and then we look for the ones we look for. These ones we see increase the strength of the ones around them. That gives we like an edge takes and we can see which we get there .

So the the key here is that it's symmetrical and we can see the image becomes asymmetrical as well so we got like that kind of feeling that it's standing out towards we. And that's what we get when we have like minuses here and plus here again this is very this is getting a bit technical now but at least we can get some kind of intuitive and Lissa's go quickly through them again. So there's sharpen ,blur ,edgin and there's an edge detect there's and boss as so as we can see these are great examples of the same image but we're getting feature maps.

Figure 3.9 : Edge Detect Feature

So we use different feature detectors to get different feature maps of the same image and therefore now we have lots of the last of this versions of this image where in each one we've tried to detect certain things in these terms they're not applicable to us. Their second boss is probably not applicable to us in terms of convolutional neural networks but age detect that's important.

We want to detect the edges, edge enhance probably not blur sharpen so certain things like edgy text. Probably the most important one for our type of work. And in terms of understanding computers they will decide for themselves or neural networks will decide for itself what's important what's not and it probably won't be even recognizable to the human eye. You won't be able to understand what those features mean.

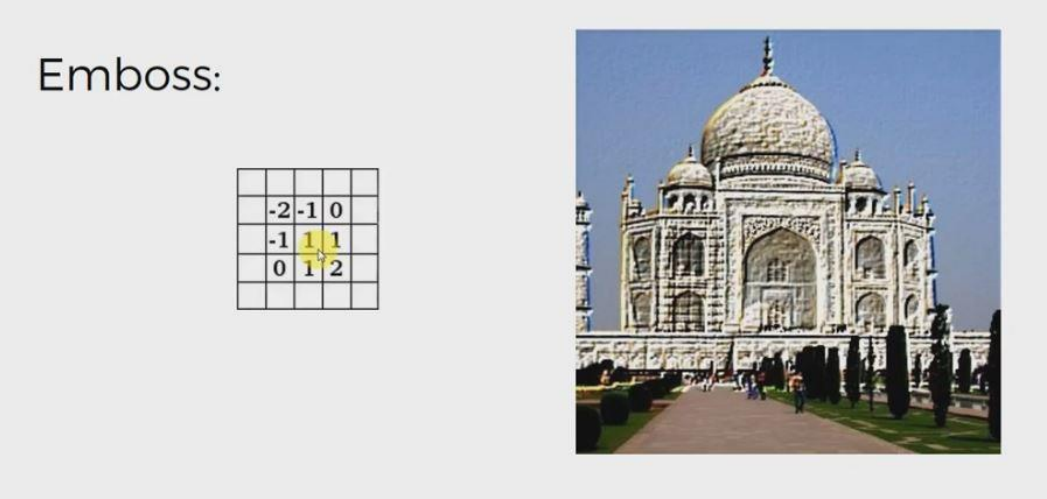


Figure 3.10 : Emboss Feature

But the computer will decide and that's the beauty of neural networks that they can process so many different things and understand without even having that intuition or without having that explanation why they will understand which features are important to them whether we have a name for them or not that , that's a whole that's an irrelevant question for the artificial neural network.

Here's a image of Geoffrey Hinton from Geoffrey Hinton passed through one of these filters.

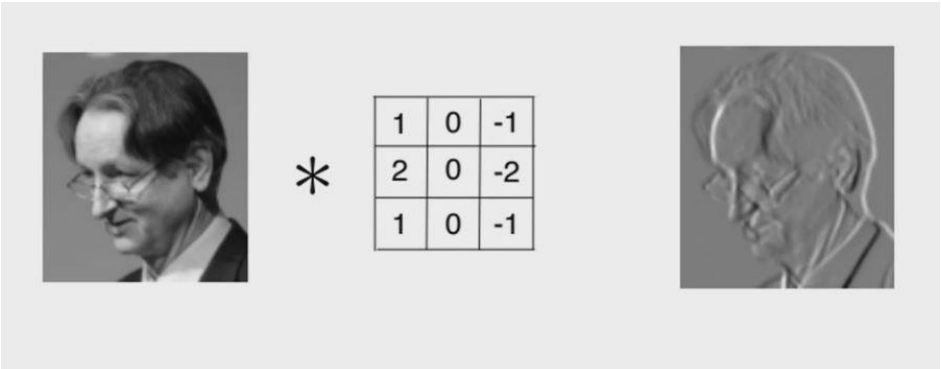


Figure 3.11 : Geoffery Hinton - Image Filters

In convolution the primary purpose of we can evolution is to find features in your image using the feature detector put them into a feature map and by having them in a future map it still preserves the spatial relationships between pixels which is very important for us to we know because if they are completely jumbled up then we've we've lost the pattern. And at the same time it's important to understand that most of the time the features a neural network will detect and use to recognize certain images and Klaas's will mean nothing to humans but nevertheless they work. And that's what convolution is.

3.3.2 Step 1(b) : ReLu Layer

There is an additional step on top of our convolution step. So it's not a separate big step it's a small step in step one be basically.

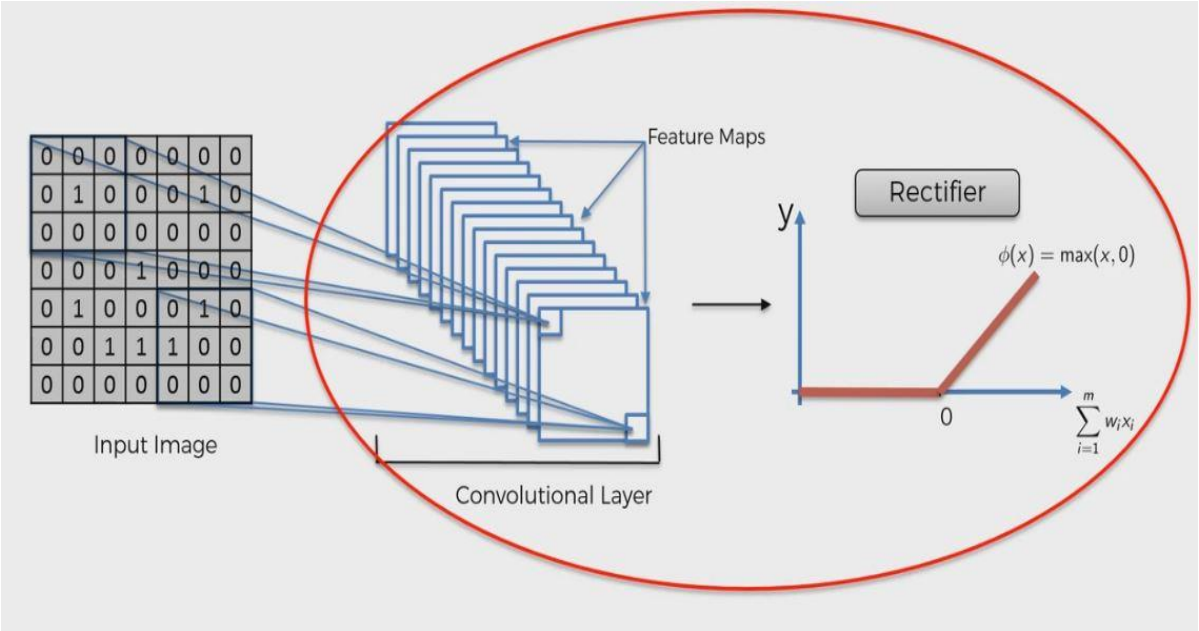


Figure 3.12 : Applying ReLu Layer

Well we have our input image we have all convolutional there which we've discussed and then on top of that we're going to apply. Our favorite rectifier function and we're familiar with the rectifier function from the previous section on artificial neural networks and in our So sometimes authors or instructors separate the convolution and direct fire as two separate steps in our examples which is going to consider them the just one big step for second evolution then the rectifier.

And the reason why we're applying the rectifier is because we want to increase non-linearity in our image or in our network and our commercial neural network and our fire acts as that filter or access function which breaks up and we arity and the reason why we want to increase nonlinearity in our network is because images themselves are highly non-linear especially if we're recognizing different objects next to each other or just on background and stuff like that like the image is going to have lots of nonlinear elements and the transition between pixels adjacent pixels is often would be nonlinear.

That's we know because its borders is different colors is different. There's different elements in your images and but at the same time when we're applying a mathematical operation such as convolution we know and running this feature detection to create our feature maps we risk that we might create something linear and therefore we need to break up the narrative. So let's have a look at an example here is a image and original image. Now when we apply a feature detection detector to this

image we get something like this. So we can see here that black is negative white is positive value as well.

Figure 3.13 : Experiment Image

When we apply a feature detector to a like a proper image which has not just zeros and ones but has lots of different values and we apply as we saw previously we Texas can have negative values in themselves sometimes we'll get negative values. And here are the black ones are negative white ones are positive. And what a rectified linear unit function does is it removes all the black rights in anything below zero it turns into zero. And so from this it turns into this right. And so it's it's pretty hard to see what exactly is the benefit in terms of in terms of breaking up linearity.

Figure 3.14 : Positives & Negatives

Figure 3.15 : Only Negatives

I'll try to explain. I'll try to show an example on this image but at the end of the day it's this is a very mathematical concept and would have to go into a lot of math to really explain what is going on. But let's let's try let's have a look. So for instance let's look at this. This building here. So this is a building on its own. Then we can see this shadow. This black part this shadow over here well we see that it's white the reflection of the light and then it's a gray and then it gets darker and then it gets darker again. So and when we take it out we take out that black spot. So think of it in terms of linearity right. So the it looks like when we go from white to gray the next step would be black. Right the next up would be black it's it's a linear progression from bright to dark and therefore this is kind of like a linear situation. When we take out the black we break up the linearity.

Let's try another one. Let's have a look here. And at the same time it's still that same building right. It's not it's not like we or your like it's not like we're blending two buildings into each other but that is secondary.

The main point is breaking up the linearity. So let's have a look here same thing. So we see white gray black gray white. And when we break it up we don't have that anymore right. You don't have that progression the gradual progression that we just have like an abrupt change.

And that helps introduce non-linearity into your image. So it's a very rough explanation very kind of like on or on the fingers explanation rather than technical but hopefully it kind of helps we understand a bit better what we're talking about here. So here again we can see white gray is a better example even to bright darker, darker , darker and darker. So this part looks like it's thinner than we break it up like that again so this is a very rough explanation. It's not absolutely perfect but at least it gives we some idea of what's going on. But if we'd like to learn more there's a good paper as always there's always a paper.

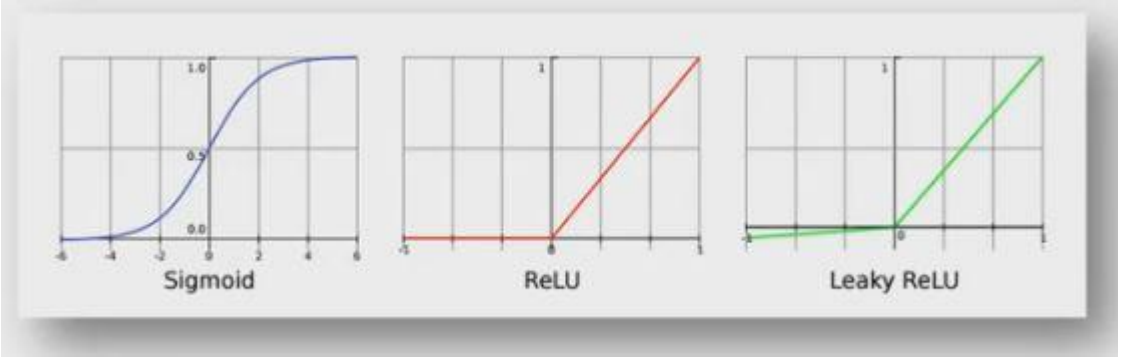


Figure 3.16 : Sigmoid & ReLu Layers

This one is by CCJ corps from the University of California and it's called Understanding convolutional neural networks who have a mathematical model. And basically they're He answers to questions and we need to just look at the first one. And the question is why is not a nonlinear activation function is essential at the filter

output of all intermediate layers. So that kind of explains it in a bit more detail both in terms of intuition and mostly in terms of mathematics.

So that's an interesting paper where we can get some more additional information on this topic. And if we really want to dig in and explore some some cool stuff here. Then there's another paper that we might be interested in. It's called delving deeper into rectifier surpassing human level level performance on image and that classification.

And here the author is combing hair and others from Microsoft Research. They propose a different type of rectified leaner unit function. They propose the parametric rectified function which we see here on the right. And they argue that it delivers better results without sacrificing performance. So interesting read if we'd like to get a bit more into this topic.

3.3.3 Step 2 : Max Pooling

We have to make sure that our neural network has a property called spatial invariance meaning that it doesn't care where the features are again not not so much as itch which part of the image because we're we've kind of taken that into consideration with our map we are poor with our convolutional there but it doesn't have to care if the features are a bit tilted if the features are a bit different in texture if the features are a bit closer of features or a bit further apart relative to relative to each other. So if the feature itself is a bit distorted our neural network has to have some level of flexibility to be able to still find that feature. And that is what pooling is all about.

So let's have a look at how pooling works. Here's our feature map so we've already done our convolution and we've completed that part and now we're working with the convolutional there. Now we're going to apply pooling.

So how does it work. We're going to be applying back pooling. There's several different types of play complies mean pooling Max pooling some pooling and will comment on those towards the end of the story. But for now we're just applying Max pooling so we take a box of two by two pixels like that and again it doesn't have to be two by two we can choose any size of box and again will comment on that towards and is Tauriel and we place it in the top left hand corner and we find the maximum value in that box and then we record only that value and we disregard the other three. So in your box we have four values we just disregard three we only keep one the maximum which is one in this case. Then we move your box to the right by stride we select the stride once again. So here we slide to stride of two and that's what we normally psyched we can say like the stride of one we can select.

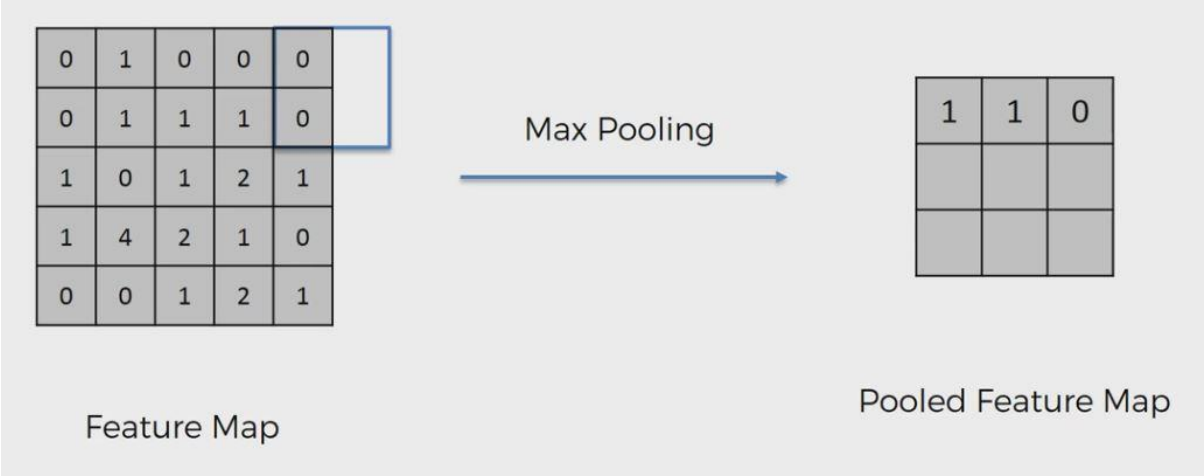


Figure 3.17 : Pooled Featured Map

So there are overlapping boxes we can select any kind of strike that we like even three if we want but we're selecting a stride of two here and that's what is commonly used. And then we repeat the repeat the process we record that maxim here if we cross over and doesn't matter we just keep continue doing what we're doing. So we still record the maximum here 0 here the maximum is four. Here are the maximums to here the maximum is 1 0 1 or 2 and then 1. So as we can see a few things happened. First of all we still were able to preserve the features right. The maximum numbers they represent because we know how the convolutional Layers works.

We know that the maximal or the large numbers in your feature map they represent where we actually found the closest similarity to a feature. But by then pooling these features we are first of all getting rid of 75 percent of the information that is not the feature which is which is not the important things that we're looking out for because we're just really three pixels out of four. So we're only getting 25 percent. And and then also because we are taking the maximum of the pixels that we or the values that we have we are therefore accounting for any distortion.

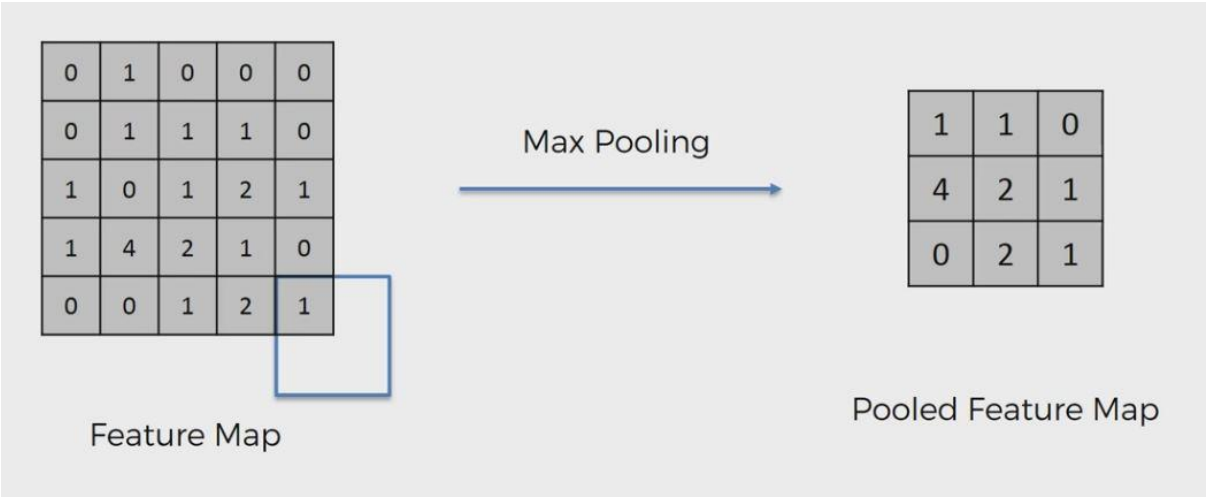


Figure 3.18: Max Pooling

Then when we are doing the pooling we're still going to get the same pool feature map and that's kind of the principle behind it. It's a very rough explanation again intuitive explanation but that's the point of pooling that we're still being able to preserve the features and moreover account for their possible spatial or textural or other kind of distortions. And in addition to all of that we are reducing the size so there's another benefit.

So we've got we're preserving the features we're introducing spatial invariants we're reducing the size by 75 percent which is huge which is really going to help us in terms of processing. And moreover another benefit of pooling is we are reducing the number of parameters so we're reducing again by 75 percent or reducing number of parameters that are going to go into our final Layers of the neural network and therefore we're preventing overfitting.

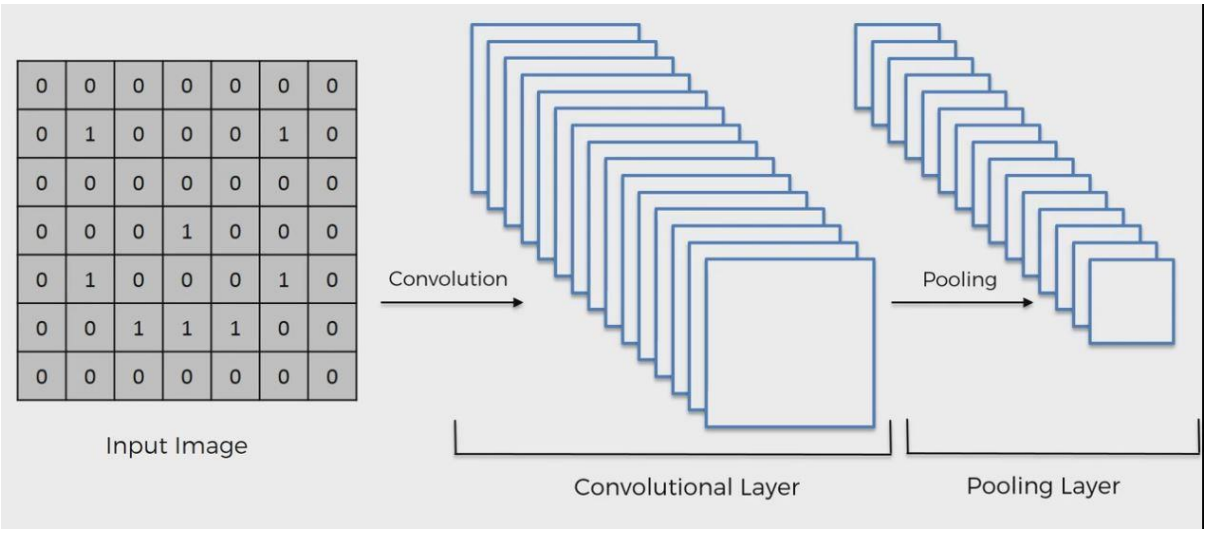


Figure 3.19: Pooling Layer

It is a very important benefit of pooling that we're removing information and that is a good thing. That is a good thing because that way our model won't be able to over fit onto that information because especially because that information is not well and remember like at the very start we're talking about even for human as humans it's important to see exactly the features rather than all this other noise that is coming into our eyes. Well same thing for neural networks they by disregarding the unnecessary non-important formation we're helping with preventing of overfitting. So there we go that is what pooling is about. And the question here is of course why WiMax pooling right there's lots of different types of pooling and a wide wide stride of too wide a size of two by two pixels lots of all these things. And on that note I'd like to introduce we to this lovely research paper called evaluation of pooling operations in convolutional architectures for object recognition by Dominic Scherrer from University of Bonn.

So in our squarer taking the maximum value there's a concept called Mean pooling or some pooling some pooling as we just some of these values up average pooling or mean pooling we take the average value out of all of these and subsampling is kind of

like a generalization of men pooling. It's it's a more kind of generalized approach to taking the average of of these values. But otherwise just think of it as average pooling

So there is our input image. Then we applied the convolution operation and we got the conclusion. And now to each of those feature maps that we get We've applied the Pooling Layer. So we've got we've done these two steps evolution and pooling and now we're going to do something very fun something exciting. We're going to experiment with this so this is a screenshot I took from a tool created by Adam Harley from way back when he was at Ryerson University of computer science and now he's at Carnegie Mellon

It's as it's just hard to find it through Google because there's no text here as we were just this year. I'll see start Reierson dossier and this stuff. And basically this is exactly what we're doing but visualize So here we need to draw a number so say I draw number four and this tool will put the number four here.

In our first step then this is the convolution step. And this is the pooling step and also pooling by the way is also called downsampling. So pooling and downsampling are the same things. So we can see it's applied convolution then it's applied pooling and we can see how it exactly works. You can see what kind of convolutions that it has applied or what kind of filters it is applied what they look like.

What features is looking out for. And then it's applied pooling so it's reducing the size and we can see here that this is important. So we can see that this is the convolved image and this is the puled image and we can still see the same features is just less information but same features right features are preserved. That's the important part. And moreover if we know if all four was a bit too kind of like rotated a bit to the side it would still be able to pick up very similar pool Lares. And then after that it's got more letters we haven't talked about that yet. So then he's got another convolutional a convolutional lair here which we actually won't have. And then he has another poor lair but he's basically just repeating that same process. And then after that this is what we're going to be talking further down in the course. He's got the fully connected layers and so on. But we can definitely play around with that.

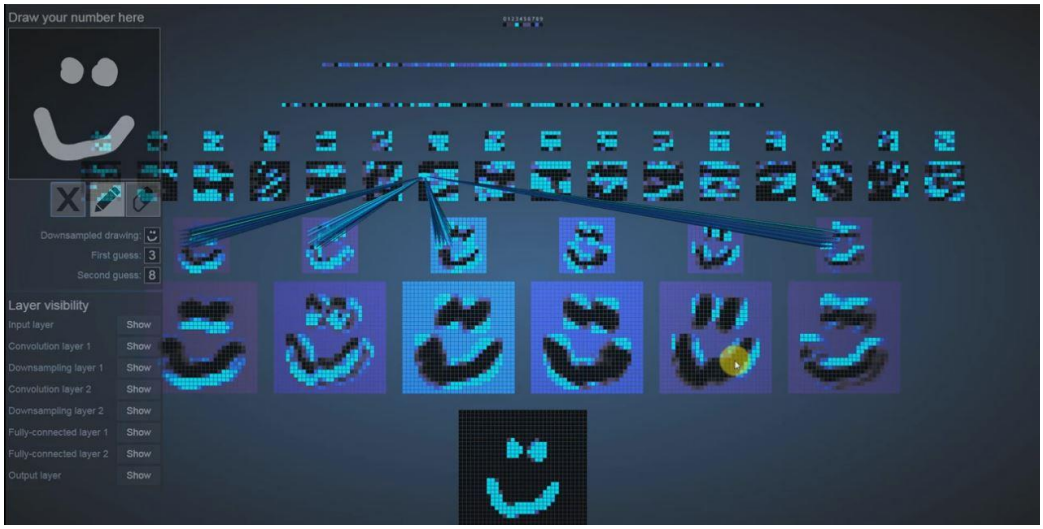


Figure 3.20 : An Online feature detector

It shows we where the feature detector was to pick up that pixel so we can see where those pixels are coming from and also so we can see how the filter was kind of like going through the image exactly how we talked about and of course and here we can see we can see the pooling we can see that the pooling is done with the pooling is done with a little square size of two by two .

3.3.4 Step 3 : Flattening

All right so we so far we've got the Pooling Layer ,Pooled feature map and that is after we apply the convolution operation to our image and then we apply pooling to the result of the collision which the involved image. And so what are we going to do if this pooled feature map. Well we're going to take it and we're going to flatten it into a column.

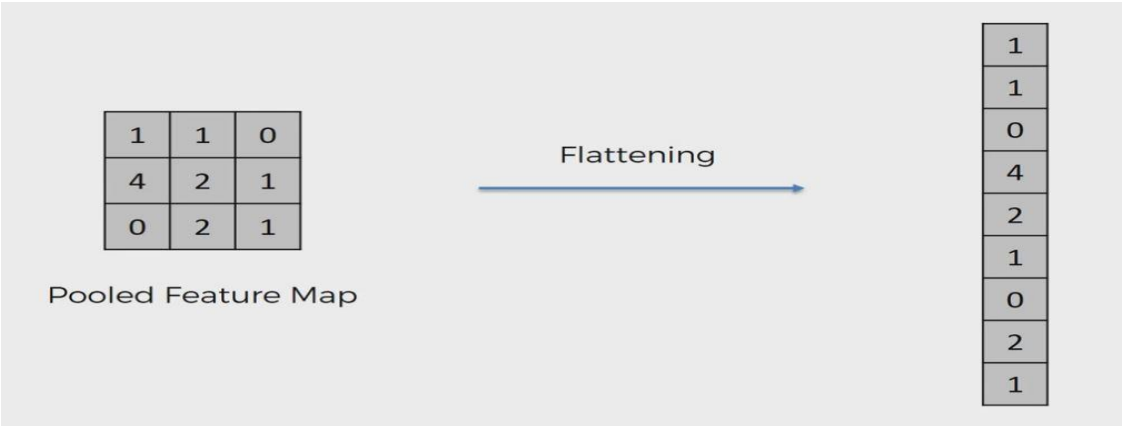


Figure 3.21 : Flattening

So basically just take the numbers row by row and put them into this one long column. And the reason for that is because we want to later input this into an artificial neural network for further processing. So this is what it looks like when we have many pooling layers or we have the pooling levers with many puled feature maps and

then we flatten them so we put them into this one long column sequentially one after the other and we get one huge vector of inputs for an artificial neural network. And so to sum all this up we've got an input image .

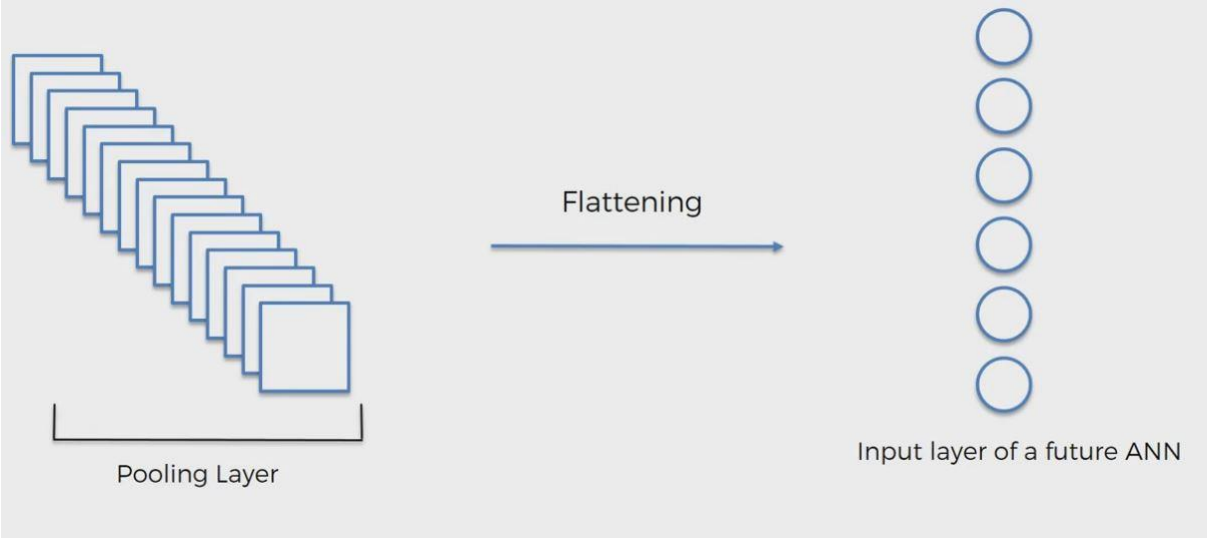


Figure 3.22 : Flattening as an Input to ANN

We apply a convolutional there and let's not forget the reals or rectified rectified linear units function that we apply after the revolution there as well. And then we apply pooling and then we flatten everything into a long vector which will be our input layer for an artificial neural network .

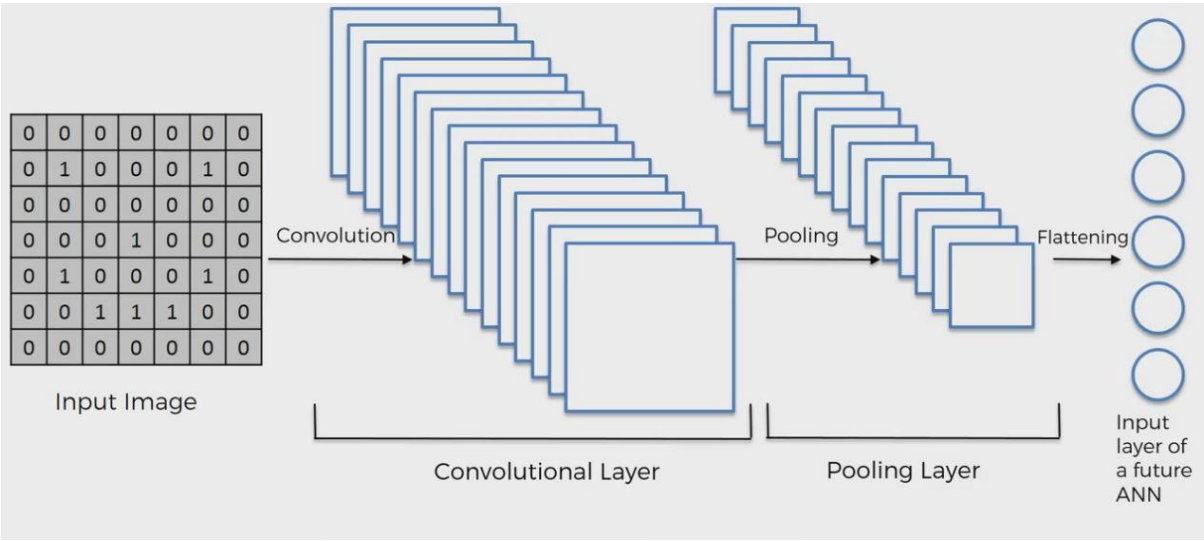


Figure 3.23 : Flattening as Input Layer of a Future Neuron

3.3.5 Step 4 : Full Connection

We're finally at STEP Four before full connection. So what is this step all about. Well in this step we're adding a whole artificial neural network to our convolutional neural

network so to all of the things that we've done so far which are convolution pooling and flattening. Now we're adding a whole new.

And then on the back of that how intense is that. That is just that is something that is definitely something. And so here we've got the input layer we've got a fully connected plan. I'll put there and by the way the fully connected Layer in the artificial neural networks we used to call them hidden layers and here we're calling them fully connected because they are hidden lairs but at the same time they're a more specific type of fiddlers that are fully connected in artificial neural networks hidden letters don't have to be fully connected.

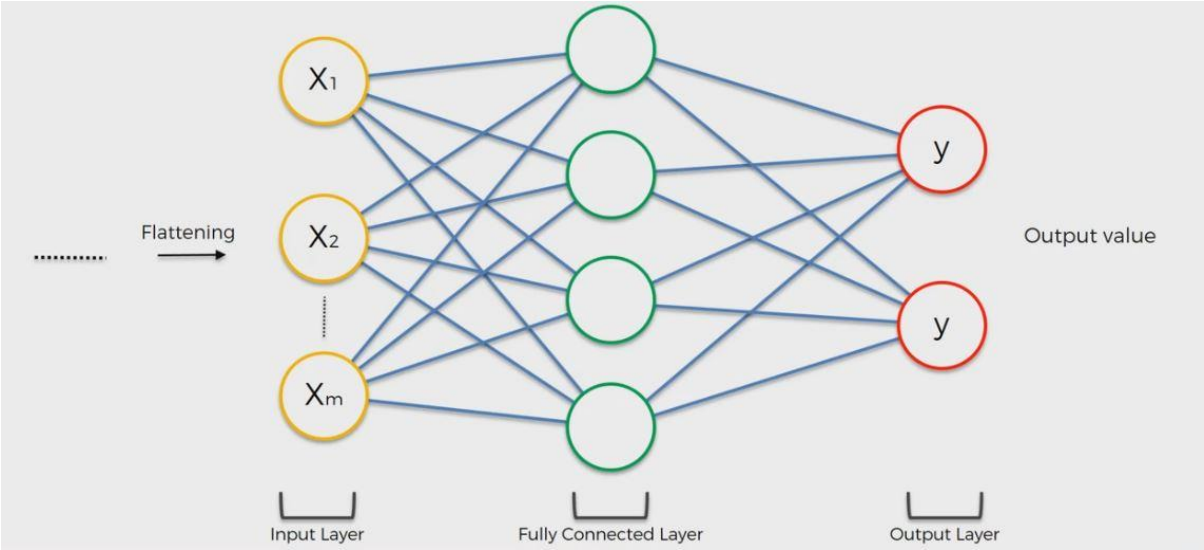


Figure 3.24 : Fully Connected Net

Whereas in convolutional neural networks we're going to be using fully connected letters and that's why they're generally called fully connected Layers. And so basically that whole column or vector of outputs that we have after the flattening we're passing it into the input learned here we've got a very simplified example just for illustration purposes.And what the main purpose of the artificial neural network is is to combine our features into more attributes that predict the classes even better.

So we already in our vector of outputs in the Flatten of the flattened result from what we've really done we have some features encoded in the numbers in that vector and they can already do probably a pretty good job at predicting what's Class we're looking at whether it's a sad or a happy or whether it's a anger or a disgust and so on. But at the same time we know that we have this structure called artificial neural network which is designed which which has a purpose of dealing with attributes and coming out or dealing with features and coming up with new attributes and combining attributes together to even better predict things that we're trying to predict and we know that from the previous parts so why not leverage that. And that's exactly what the plan here is. So how about we pass on those values into an artificial neural

network and let it even further optimize everything that we're doing. And so that's what we're going to be doing. But let's look at a more realistic example because this one is a bit too simple. So here we've got a better looking artificial neural network where we have five attributes on the inputs that we have in the first unless we have six neurons in the second or in the second fully connected Larry have eight neurons and then we have two outputs one for sad and one for happy. And so an important thing to talk of for us to talk about here is that why do we have two outputs.

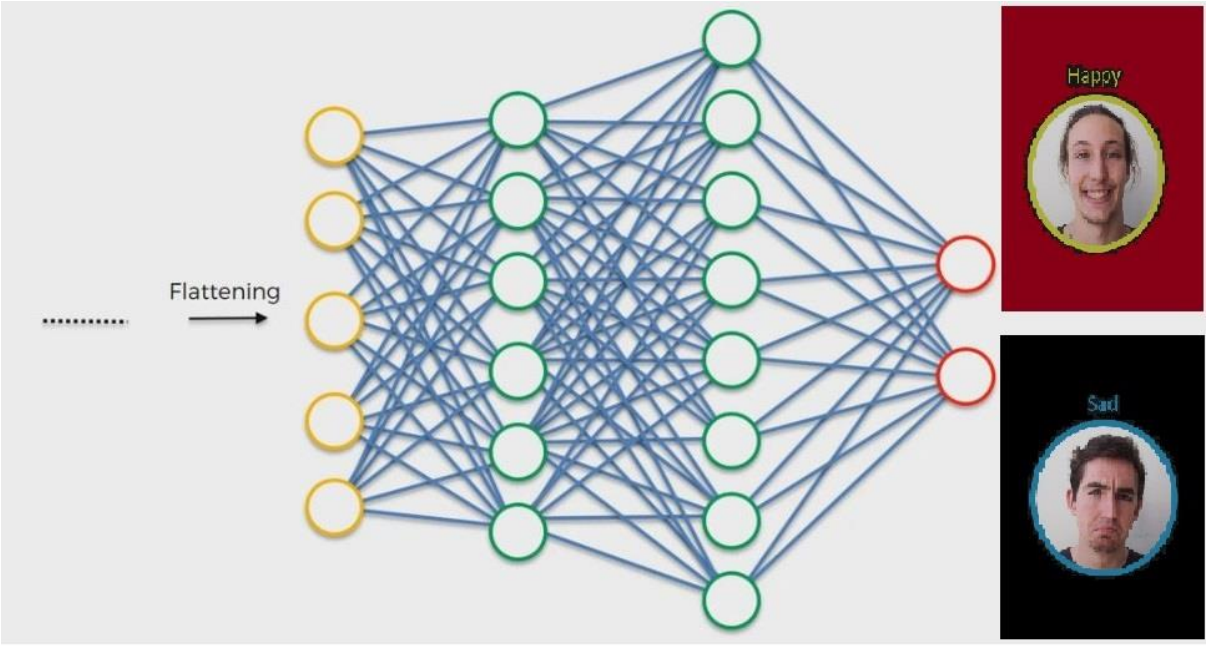


Figure 3.25 : Binary Output - Either Happy or Sad

We're kind of used to having only one output in our artificial neural networks Well one output is for kind of when we're predicting a numerical value when we print when we're running a regression type of problem. But when we're doing classify happy on we need an output Proclus except for the exception is when we have just two clusters like we have two classes here sad and happy and we could have just done one output and made it a binary output and said One(1) is a sad face and zeros(0) a happy face and that would have worked totally fine. But at the same time if we have more than two categories for instance sad , happy and anger then we have to have a neuron per every category and that's why we're going to practice with two categories in this example so that we know what to expect if we ever have ***more than two categories***. And so what's going to be happening here.

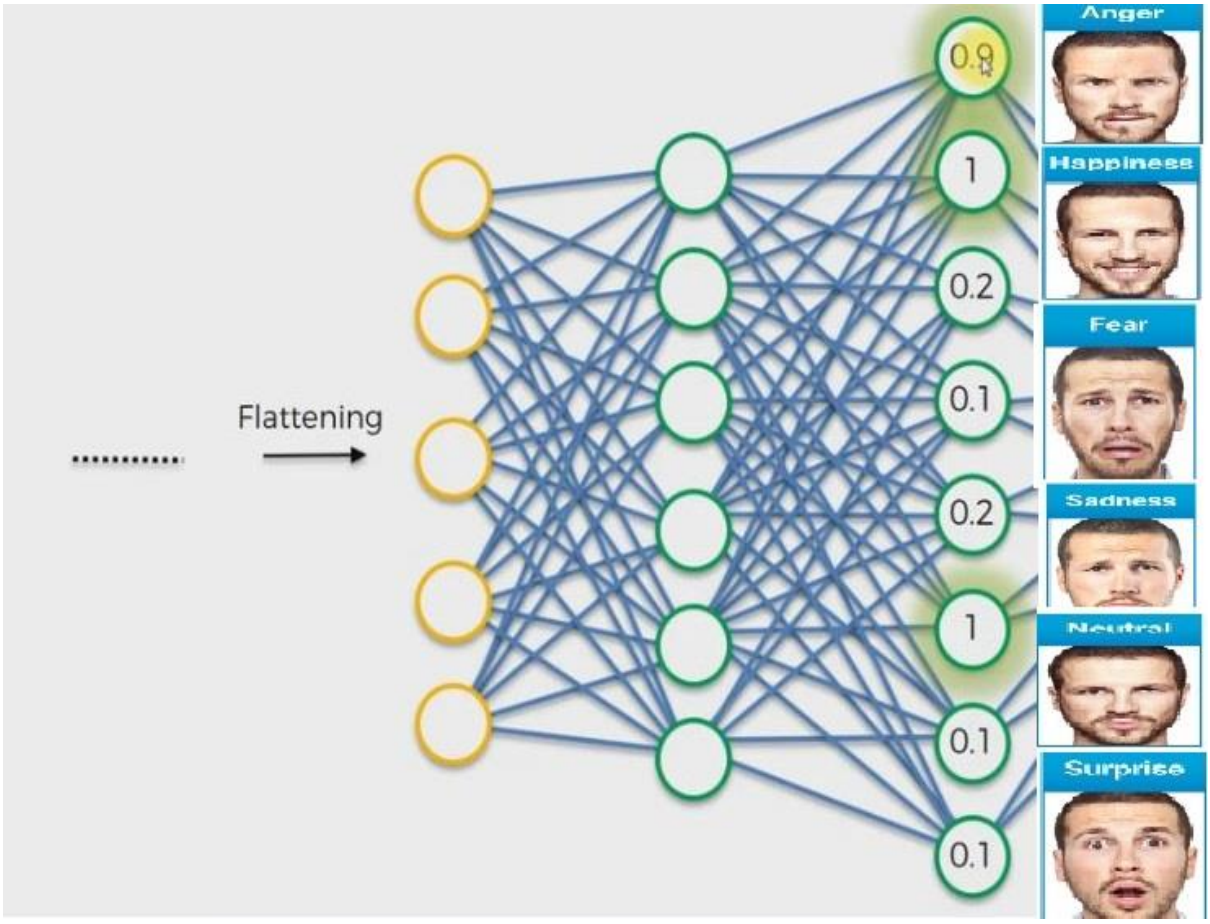


Figure 3.26 : Multiple Output Neurons as Different Emotions

So we've already done all the groundwork we've done the convolution we've done the pooling and the flattening and now the information is going to go through the artificial neural network so let's have a look at how the other all happens. There is information going through from the very start from the moment when the image is processed and kind convoluted convolved then pooled ,flattened and then through the artificial neural network all four steps and then a prediction is made and we'll see how this happens in a moment will be very very interesting. But for now let's just say a prediction is made. And for instance 80 percent that it's a sad. But it turns out to be a happy and then an error is calculated. Well what we used to call accosts cost function in a artificial neural network and we used mean square error there or in-common illusional neural networks. It's called a loss function and we use a cross entropy function for that. And we'll talk about cross entropy and mean squared errors.

For our knowledge we say we have a lost type of function which tells us how well our network is performing and we're trying to optimize it or minimize that function to optimize our network. So the error is calculated and then it's back propagated through the network just like we had in artificial neural networks is back propagated and the some things are adjusted in the network to help optimize the performance and the things that are adjusted are as usual the weights in the artificial neural network are part of those the blue lines that we see here the Cynapsus. Then also another thing that

is adjusted is the feature detectors so we know that we're looking for features but what if we're looking for the wrong features. What if this didn't work out because the features are incorrect and so the feature detectors those remember those little matrices that we had.

That's the three by three matrices. They are adjusted so that maybe next time it'll be better and let's see what happens. And but of course it's all done with a lot of science in the background of a lot of math and it's all done through a gradient gradient descent of back propagation. So it's all it's all not just random perturbations it's actually very thought through how it's done. But nevertheless the feature detectors are adjusted the weights are adjusted and this whole process happens again and then again the errors back propagate. And this keeps going on and on and on.

And that's how our network is optimized that's how our network trains on the data. So the important thing here is that the data goes through the whole area from the very start to the very end. Then the error is compared so the error is calculated and then is back propagated. So same story as with artificial neural networks just a bit longer because of that whole for the first three steps that we already had. And now let's have a look at the interesting part the really interesting part how do these two classes work because Or how do these two output neurons work because before we've always kind of had one output neuron what happens when we have two. How does how does this situation of classifihappyion or images play out. Well let's start with the top neuron first going to start with the sad. How do we the main purpose what we need to do first is we need to understand what weights to assign to all of these syllabuses that connect to the sad so that we know which of the previous neurons are actually important for the sad and let's see how that is done. So let's say hypothetically we've got these numbers in our previous layer of previous fully connected. In the final we get fully connected layer.

And again these numbers can be absolutely anything. They don't have to be that they can be any numbers but just for argument's sake we're going to agree that we are looking specifically at numbers between 0 and 1. So it's easier for us to argue these things and understand and one means that that neuron was very confident that it found this feature and zero is going to mean that that neuron didn't find a feature is looking for so because at the end of the day these neurons like anything else on this from on this left side is just looking at features at an image. This is already very very process.

But still it's detecting a certain feature or a combination of features on the image right before we can evolve step. We had kind of recognizable features in the pool set they're less recognizable than they become even less recognizable in the flattened image. And then they get combine and so on. But nevertheless this we're talking about here certain features that are present image or their combination. So and one which

has been passed and this is important has been passed to both the sad and the happy a the same time to both the output neurons.

So one means that for us for our argument it means that this this neuron has is firing up its It's really rapidly detecting that feature that we know might be an eyebrow it might be detecting this eyebrow for again for simplicity sake is detecting this eyebrow. And is communicate that to the sad run to the happy neuron saying I can see my eyebrow I can see my eyebrow. And then it's up to the sad and the happy neuron to understand what that means for them. And so in this case which neurons are firing up these three neurons are firing up the eyebrow and that say the nose is saying I can see I can see a big nose and I can see floppy ears.

So it and it's saying that to the sad and to the happy and then what the sad. And then what happens is we know that this is a sad. So the sad neuron knows that the answer is it is actually a sad because at the end we're comparing to the picture or to the label on the picture and when another sad. So basically the sad neuron is going to say Aha. So I should be triggered in this case. So these are neurons they're telling this signal that they're sending to both to me to the sad and the happy is actually a indihappyion for me that it is a sad. And throughout these lots and lots and lots of iterations of this happens many times the sad will learn that these neurons do indeed fire up when the feature belongs to a sad. On the other hand the happy neuron will know that it's not a happy and it will know that this feature is firing up and this neuron is telling me it can see floppy ears . But at the same time it's not a happy. So basically to me that's a signal that I should ignore this neuron like and the more that happens the more the happy neuron is going to ignore this neuron about the floppy ears.

And so basically that that's how through lots and lots of iterations if this happens often. So this is just one example but if this happens often maybe a one maybe 0.8 0.9 maybe sometimes it won't fire but overall on average this neuron is lighting up very often when it is indeed a sad the sad neuron will start attributing higher importance to this neuron. And so there we go. That's that's how we're going to signify it. We're going to say that these three neurons through this iterative process with me with many many many many samples and many many a remember so a sample is a row in your data set and Apoc is when we go through your whole dataset again and again and again there are lots and lots of iterations. This sad neuron learned that this eyebrowed neuron and this big nose neuron and this floppy ear neuron they all seem to really contribute very well to the classifihappyion of what it's looking for and which is a sad. So that's how it works.

And again these ears and nose and eyebrows those are very very approximate or like very far fetched examples because by this stage in this whole convolution conventional neural network it is completely unrecognizable what they're looking for but at the same time it is something in the features of sad or happy or whatever we classify it. And then so let's move on to the next.

Now we're going to look at the happy neuron but these We're going to remember that these weights are we know how we've sorted out the sad. So the sad is kind of like pretty much ignoring all these other neurons one two three four or five but it's really paying attention to what these three neurons are saying. Now what is the happy's listening to. Well whenever it is actually a happy. This is this is an example of a situation when it's actually a happy.

So we'll see that this these three neurons 0.9 0.9 and one they're saying something they're saying something to both the sad and the happy. And this is again important remember so this output signal goes both ways it's the same right. It's saying one to the sad saying to the happy but then it's up to the sad to the happy to decide whether to take into account that signal and learn from it or not.

And both the sad and the happy can see that this is a photo I should of put a photo of a happy here but basically imagine a photo of a happy both a sad and the happy can see that this is actually a happy. So basically the sad is like OK so these whiskers and these pointy triangle ears and this small size yes or or maybe this type we know how happys have these things in their eyes their eyes are like little They're not circles or lines or something like happy eyes.

Basically these happy eyes they're definitely not working for me. They're not helping me I'll predict because every time these neurons light up the prediction is not what I'm looking for. On the other hand the happy is like hmm that's interesting. Every time these this one lights up it's more most of the time it lights up. It matches my expectation it matches what I'm looking for.

You know this one useless to me because he's not actually we know like he's he's not even lighting up it's a happy but it's he's not lighting up so the opposite is happening. And this one is well it's a cad but he's not letting up so I'm not gonna listen to him. But this one when he went what was this the eyes the happy eyes light up we can see I can see that it's a happy. It matches most of the time so I'm going to learn from that and I'm going to listen to these three guys more often than not.

And so basically the happy is listening to these three and it's ignoring the other five and that is how these final neurons learn which neurons in the final fully connected layers to listen to the output neurons learn which of the fully which are the final fully connected.

There are neurons to listen to. And that's how they understand. Basically that's how the features are propagated through the network and conveyed to the output. And so even though these features of course don't have that much meaning to them like floppy ears or whiskers.

At the same time they do have some distinctive they are a distinctive feature of that specific class and that's how the network is trained because we also during remember

during the back propagation process we also adjust the feature detectors so if a feature is useless to the output it's going to it's going to probably be disregarded because this doesn't happen to one or two stories it happens through thousands and thousands of iterations. So with time a feature that is useless to the network is going to be disregarded and replaced with feature is useful and so at the end of the day in this final layer of neurons we are likely to have lots of features or combinations of features from the image that are indeed representative or descriptive of sad and happy.

And so then once your network is trained up then we this is how it's applied. So this is the next step like we were trained up our network will this happen. Let's see what happens when the this network is applied. So let's say we pass on an image of a sad. The values are propagated through a network we get certain values. And so this time the sad and the happy neurons don't know they don't have the image of the sad here they don't know that it's a sad or a happy. They have no idea what it is but they have learned to listen to what is being shown here.

They have learned to listen to sad and listens to these three neurons a happy neuron listens to these three. And so the sad neuron looks at one two three and says aha these are pretty high. So my probability is going to be high that is a sad the happy neuron looks at these three and says OK these this one is pretty high but these are pretty low. So my probability is going to be 0.05. And and then and that's. And that's where we get your prediction. So then your first choice for this neural network is sad.

Second choice is happy and that's pretty much it. So the answer is sad and same thing happens when we pass an image of a happy. You get new values and we can see that even though this one's high these ones are low. And for the happy This was high this was high and this one's a bit low. So the probability here might not be as great as previously but still we can see that it's a happy of 79 percent. And so therefore the neural network is going to vote that it's a happy. And so basically all the neural networks going to conclude that it's a happy. Voting is a term that is used for these guys so these neurons in the final fully connected Layers they get to vote. And these are their votes. And again we are just for argument's sake putting values between 0 and 1 here. These could be any values but they get to vote and then these weights are the importance of their vote. So this is these are these purple weights are how the sad neuron views their votes.

How much importance is it assigns to these neurons and those votes. And this is how much importance the happy's neuron up size to these votes the votes of these neurons and so these neurons vote the sad and the happy based on their learned the weights they decide who to listen to and then they make their predictions and then hold neural network concludes that this is in this case a happy and then that's And then that's your conclusion and that's how we get images like this where we have a cheetah and then we have a cheetah claws who we know like a high probability So this is we know the probability that the network has predicted. And these are laws but these still exist because they're still kind of like a small chance the other neurons are also listening to

their voters and they're saying oh maybe it's actually a disgust or an anger face. Very very probable. I hear scissors. You know this one one but hand-glass was a very close second and in fivepence stethoscope because we could see like this guy this this neuron the scissors neuron the output series neuron listened to its voters and it had the predominant vote overall. But then the hand-glass had a good outcome as well.

So there we go that's how the full connection works and how this is all this all plays out together.

3.3.6 Summarizing the Steps

We started with an input image to which we applied multiple different feature detectors or also called filters to create these feature maps. And this comprises our convolutional layer.

Then on top of that crucial Layers we applied the relu or rectified linear unit to remove any clarity or increase non-linearity in our images.Then we applied a pooling lair to our convolutional layer. So from every single feature map we created a puled feature map and basically the pooling Layers has lots of advantages. The main purpose of the pooling Lair is to make sure that we have a special spatial invariants in our images. So basically if something tilts or twists or is a bit different to the ideal scenario then we can still pick up that feature plus pooling significantly reduces the size of our images.

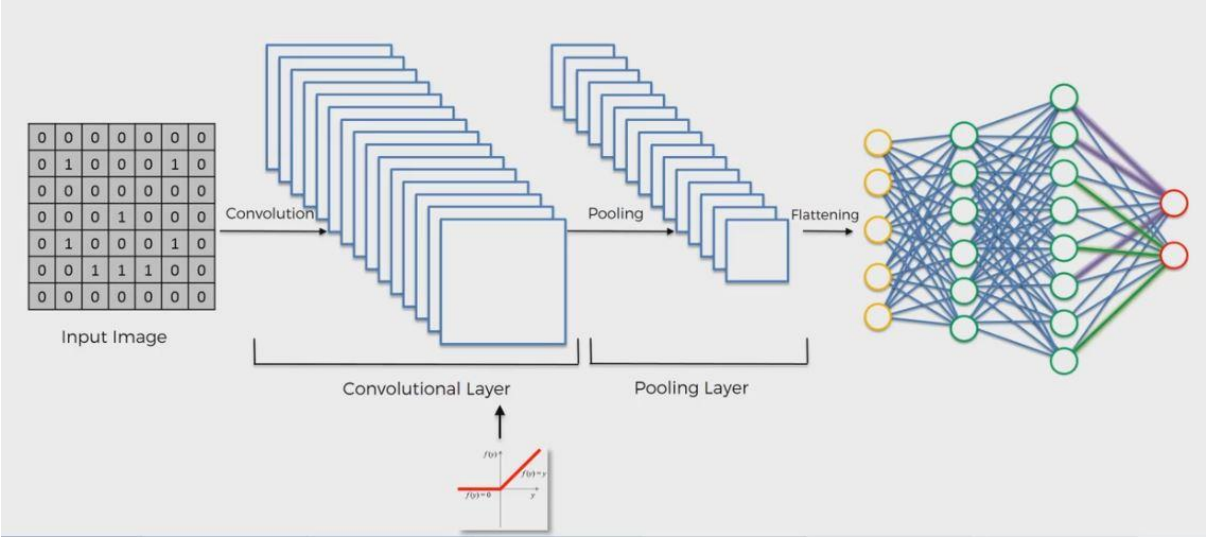


Figure 3.27 : Full Connection

Also pooling helps with avoiding any kind of overfitting of our data or overall model to the data because it just simply gets rid of a lot of that data. But at the same time pooling preserves the main features that we're after just because the way instruction

and the pooling were used was Max pooling. Then we flattened all of the pooled images into one long vector or column of all of these values and we put that into an artificial neural network and that was step flattening. And Step four is a fully connected artificial neural network where all of these features are processed through a network and then we have this final layers final fully connected layer which performs the voting towards the classes that we're after and then all of this is trained through a forward propagation and back propagation process. Lots of lots of iterations and at parks and in the end we have a very well defined neural network.

Another important thing is not only the weights are trained in artificial neurons work part but also the feature detectors are trained and adjusted in that same ingredient decent process and that allows us to come up with the best feature maps. And in the end we get a fully trained convolutional neural network which can recognize images and classify them. So there we go. That's how convolutional neural networks work.

3.3.7 Technology Used

Anaconda

Anaconda Cloud is where data scientists share their work. You can search and download popular Python and R packages and notebooks to jumpstart your data science work. You can also store your packages, notebooks and environments in Anaconda Cloud and share them with your team.

Spyder

Spyder is a tool which works as a framework in Anaconda for running Python Scripts .

Tensorflow

TensorFlow is an open source programming library for numerical calculation utilizing information stream charts. Hubs in the chart speak to numerical operations, while the diagram edges speak to the multidimensional information exhibits (tensors) conveyed between them. The adaptable engineering enables you to convey calculation to at least one CPUs or GPUs in a desktop, server, or cell phone with a solitary API. TensorFlow was initially created by scientists and specialists taking a shot at the Google Brain Team inside Google's Machine Intelligence inquire about association for the reasons for directing machine learning and profound neural systems examine, however the framework is sufficiently general to be appropriate in a wide assortment of different spaces too.

CHAPTER 4 : CONCLUSION

In the course of the most recent decade, expanding considerations have been coordinated to the investigation of FER. For facial component extraction strategies, geometric element based techniques and appearance-based techniques, utilized for static pictures, were first explored. At that point, these facial component extraction techniques for dynamic picture groupings, including optical flow and highlight point following, were likewise researched. For outward appearance classification, six run of the mill classification strategies including HMM, ANN, BN, KNN, SVM and SRC, were additionally reviewed. Likewise, we performed FER probes the JAFFE database and the Cohn Kanade database, and introduced a similar investigation of various classification techniques in view of the extricated LBP highlights. Analysis comes about demonstrate that SRC outflanks the other utilized techniques, for example, KNN, SVM, HMM, guileless Bayes, and ANN.

Albeit broad endeavors have been given to FER and numerous current victories have been accomplished, as said above, many inquiries are as yet open. In our assessments, the accompanying a few focuses ought to be considered in future.

(1) How do people effectively distinguish facial expression?

Up until this point, mental and restorative inquires about on human recognition and comprehension have gone on for quite a while, yet it is as yet equivocal how people distinguish outward appearance. Which sorts of parameters can be utilized by people and how are they prepared?

(2) How would we recognize facial expression in true views?

Because of unobtrusive facial misshapenings, visit head developments, and equivocal and indeterminate facial movement estimations, recognizing unconstrained outward appearance in certifiable views is significantly more difficult than the acted FER broadly concentrated to date.

(3) How would we consequently take in more successful facial highlights for facial expression detection ?

It merits bringing up that the previously mentioned handdesigned highlight extraction techniques more often than not depend on manual operations with marked information. At the end of the day, these strategies are managed. What's more, these handdesigned highlights, for example, LBP and Gabor wavelets portrayal can catch low-level data of facial pictures, aside from abnormal state portrayal of facial pictures. As an as of late rose machine learning hypothesis, depends on the various leveled engineering of data preparing in the primate visual observation framework, and has demonstrated how orders of highlights can be specifically gained from unique information in an unsupervised way. The most effective method to utilize profound

learning strategies to naturally take in more compelling facial highlights is a critical course for FER.

(4) How may we incorporate facial expression investigation with different modalities?

Feeling exchanges the mental data of people, since feeling is passed on by different physiological changes, for example, changes in heart-pulsating rate, sweating degree, pulse, and so forth. Feeling is additionally communicated by full of feeling discourse, outward appearance, body signal et cetera. To advance feeling acknowledgment execution, coordinating numerous full of feeling modalities, for example, discourse, facial, physiological and lexical data, is an exceptionally dynamic subject . By and by, how to adequately coordinating heterogeneous modalities of feeling articulation to additionally enhance multimodal feeling acknowledgment execution is as yet an open inquiry.

REFERENCES

[1] A. Mehrabian, “Communication without words,” Psychol. Today, Vol. 2, pp. 535, 1968.

[2] G. Sandbach, S. Zafeiriou, M. Pantic, and L. Yin, “Static and dynamic 3D facial expression recognition: A comprehensive survey,” Image Vis. Comput., Vol. 30, no. 10, pp. 68397, Oct. 2012.

[3] Y. Tian, T. Kanade, and J. Cohn, “Facial expression analysis,” in Handbook of face recognition. Springer, 2005, pp. 24775.

[4] C.-D. Caleanu, “Face expression recognition: A brief overview of the last decade,” in IEEE 8th International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, 2013, pp. 15761.

[5] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, “Active shape models-their training and application,” Comput. Vis. Image Underst., Vol. 61, no. 1, pp. 3859, Jan. 1995. [6] Y. Chang, C. Hu, R. Feris, and M. Turk, “Manifold based analysis of facial expression,” Image Vis. Comput., Vol. 24, no. 6, pp. 60514, Jun. 2006.

[7] R. Shbib, and S. Zhou, “Facial expression analysis using active shape model,” Int. J. Signal Process. Image Process. Pattern Recognit, Vol. 8, no. 1, pp. 922, 2015.

[8] L. A. Cament, F. J. Galdames, K. W. Bowyer, and C. A. Perez, “Face recognition under pose variation with local Gabor features enhanced by active shape and statistical models,” Pattern Recognit., Vol. 48, no. 11, pp. 337184, Nov. 2015.

[9] T. F. Cootes, G. J. Edwards, and C. J. Taylor, “Active appearance models,” IEEE Trans. Pattern Anal. Mach. Intell., Vol. 23, no. 6, pp. 6815, Jun. 2011. [10] Y. Cheon, and D. Kim, “Natural facial expression recognition using differential-AAM and manifold learning,” Pattern Recognit., Vol. 42, no. 7, pp. 134050, Jul. 2009.

[11] E. Antonakos, J. Alabort-i-Medina, G. Tzimiropoulos, and S. Zafeiriou, “Hog active appearance models,” in 2014 IEEE International Conference on Image Processing (ICIP), Paris, 2014, pp. 2248.

[12] R. Anderson, B. Stenger, and R. Cipolla, “Using bounded diameter minimum spanning trees to build dense active appearance models,” Int. J. Comput. Vis., Vol. 110, no. 1, pp. 4857, Oct. 2014.

[13] Y. Chen, C. Hua, and R. Bai, “Regression-based active appearance model initialization for facial feature tracking with missing frames,” Pattern Recognit. Lett., Vol. 38, pp. 1139, Mar. 2014.

[14] D. G. Lowe, “Object recognition from local scale-invariant features,” in the Seventh IEEE International Conference on Computer vision, Kerkyra, 1999, pp. 11507.

[15] D. G. Lowe, “Distinctive image features from scaleinvariant keypoints,” Int. J. Comput. Vis., Vol. 60, no. 2, pp. 91110, Nov. 2004.

[16] S. Berretti, A. Del Bimbo, P. Pala, B. B. Amor, and D. Mohamed, “A set of selected SIFT features for 3D facial expression recognition,” in 20th International Conference on Pattern Recognition, Istanbul, Turkey, 2010, pp. 41258.

[17] H. Soyel, and H. Demirel, “Facial expression recognition based on discriminative scale invariant feature transform,” Electron. Lett., Vol. 46, no. 5, pp. 3435, Mar. 2010.

[18] Y. Li, W. Liu, X. Li, Q. Huang, and X. Li, “GA-SIFT: A new scale invariant feature transform for multispectral image using geometric algebra,” Inform. Sci., Vol. 281, pp. 55972, Oct. 2014.

[19] T. Ojala, M. Pietik€ainen, and D. Harwood, “A comparative study of texture measures with classification based on featured distributions,” Pattern Recognit., Vol. 29, no. 1, pp. 519, Jan. 1996.

[20] C. Shan, S. Gong, and P. W. McOwan, “Facial expression recognition based on Local Binary Patterns: A comprehensive study,” Image Vis. Comput., Vol. 27, no. 6, pp. 80316, May 2009.

[21] S. Zhang, X. Zhao, and B. Lei, “Facial Expression Recognition Based on Local Binary Patterns and Local Fisher Discriminant Analysis,” WSEAS Trans. Signal Process., Vol. 8, no. 1, pp. 2131, 2012.

[22] X. Zhao, and S. Zhang, “Facial expression recognition using local binary patterns and discriminant kernel locally linear embedding,” EURASIP J. Adv. Signal Process., Vol. 2012, no. 1, pp. 20, Dec. 2012.

[23] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen, “Local binary patterns and its application to facial image analysis: a survey,” IEEE Trans. Syst. Man, Cybernet. Part C: Appl. Rev., Vol. 41, no. 6, pp. 76581, Nov. 2011.

[24] G. Zhao, and M. Pietikainen, “Dynamic texture recognition using local binary patterns with an application to facial expressions,” IEEE Trans. Pattern Anal. Mach. Intell., Vol. 29, no. 6, pp. 91528, Jun. 2007.

[25] T. Jabid, M. H. Kabir, and O. Chae, “Robust facial expression recognition based on local directional pattern,” ETRI J., Vol. 32, no. 5, pp. 78494, Oct. 2010.

[26] T. Ahsan, T. Jabid, and U.-P. Chong, “Facial expression recognition using local transitional pattern on Gabor filtered facial images,” IETE Tech. Rev., Vol. 30, no. 1, pp. 4752, Jan.Feb. 2013.

[27] X. Li, Q. Ruan, Y. Jin, G. An, and R. Zhao, “Fully automatic 3D facial expression recognition using polytypic multi-block local binary patterns,” Signal Process., Vol. 108, pp. 297308, Mar. 2015.

[28] Z. Zhang, M. Lyons, M. Schuster, and S. Akamatsu, “Comparison between geometry-based and Gabor-wavelets-based facial expression recognition using multi-layer perceptron,” in Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998. Proceedings, Nara 1998, pp. 4549.

[29] S.-s. Liu, and Y.-t. Tian, “Facial expression recognition method based on Gabor wavelet features and fractional power polynomial kernel PCA,” Adv. Neural Netw.-ISNN 2010. Vol. 6064, no. Part 2, pp. 14451, 2010.

[30] W. Gu, C. Xiang, Y. Venkatesh, D. Huang, and H. Lin, “Facial expression recognition using radial encoding of local Gabor features and classifier synthesis,” Pattern Recognit., Vol. 45, no. 1, pp. 8091, Jan. 2012.

[31] E. Owusu, Y. Zhan, and Q. R. Mao, “A neural-AdaBoost based facial expression recognition system,” Expert Syst. Appl., Vol. 41, no. 7, pp. 338390, Jun. 2014.

[32] M. J. Lyons, J. Budynek, and S. Akamatsu, “Automatic classification of single facial images,” IEEE Trans. Pattern Anal. Mach. Intell., Vol. 21, no. 12, pp. 1357362, Dec. 1999.

[33] T. Kanade, Y. Tian, and J. Cohn, “Comprehensive database for facial expression analysis,” in International Conference on Face and Gesture Recognition, Grenoble, France, 2000, pp. 4653.

[34] C. Shan, S. Gong, and P. McOwan, “Facial expression recognition based on Local Binary Patterns: A comprehensive study,” Image Vis. Comput., Vol. 27, no. 6, pp. 80316, May 2009.

[35] A Krizhevsky, I Sutskever, GE Hinton “Imagenet classification with deep convolutional neural networks” 2012

[36] Matthew D Zeiler , Rob Fergus “Visualizing and Understanding Convolutional Networks” 2013

[37] Karen Simonyan, Andrew Zisserman “ Very Deep Convolutional Networks for Large-Scale Image Recognition” 2014

[38] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich “Going Deeper with Convolutions ” 2015

[39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun “ Deep Residual Learning for Image Recognition” 2015

[40] Ross Girshick “Fast R-CNN” 2015

[41] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks” 2015