Easy things are hard in Vision -
Computer Vision

(Master the fundamentals of Convolutional Neural Network-Phase 1)

Vivek Chaudhary
Deepanshu Dashora

WHY IS THIS BOOK ?

Let’s start with a question, How will you identify apples from a basket full of different kinds of fruits?

Well, if you see an apple with different shapes, sizes & colors you can easily say it is an apple. It doesn't even matter if that apple is painted or it is a toy, you will still call it an apple. If we consider this case & understand how we are thinking biologically, we are just classifying the objects based on our own previous experiences, in other words you have stored such information while learning from childhood. Definitely, If you are asked to recognize a fruit which has never been seen before, you will not be able to name it.

See closely the process of classifying or identifying the objects with respect to human perspective we have stored such information in our brain that how apple looks like,it can be of different color & if you observer we don’t give importance to which color it is majorly classifying or identifying the objects based on their shapes first.

Well, computers deal with the problems in a kind of the same way but in a different manner. For example, if we want to train a system that can recognize apples from different fruits, we will only pick apples. For that, we have to consider all kinds of apples, sizes, shapes and colors. Even after that if we missed out on something, the system will fail in that particular case.

But can we compare this with humans, even though you have not seen a pink apple in your life, if it is asked you to recognize it you will easily say it is a pink apple based on the information processed in childhood.

If we go deeper and understand the difference, we as a human always focus on shapes and structure, not colors. In the case of computers it is about how you represent it, if you only train with recognizing apples, it will fail against pink apples, since it has never seen it before.

When we often try to make smoke crazy computer vision systems, sometimes we forget to consider some really basic things and especially the actual work procedure. Consider the last time you ever came across a lesson where you were getting explained a highly complex mathematical thing in a real-life perspective with some real-time problem statements example?

Well, Even though we study everything we always lack to try it out practically, the reason is, it is not always possible. But can we make an understanding about the use case & connect it with the procedure. Well, we can make things easy.

We are coming up with a book named “**Easy things are hard in Vision - Computer Vision**”. It will tell you the intuition of Image processing by comparing it with the Mathematical & Real-life perspectives. This book will help you understand the process of feature extraction and feature processing of the images.

Did you ever consciously think, How do humans recognize objects, differentiate between objects? I am sure 95% of us don’t focus on this,unconsciously use our brain & eyes. This book will not only help you to understand the process of Image processing but make you feel it & help you to co-relate these steps with the functioning of how humans work in the same manner.

Learning is not about just remembering a few formulas or concepts, it is about grabbing the insights and preparing ourselves to use this as a tool for solving problems. When we consider this point, we need to find a way to do it, and it is not always possible since a lack of resources may just not be enough guidance.

With this book, we bring that phase in your mind where you can understand these complex & crazy concepts with a deep understanding of the actual procedure & obviously with some practical real-time use cases, which even we face.

This is part 1 of the series with respect to Computer Vision which will cover the fundamentals of neural networks from scratch to technical aspects intuitively in layman words.

The most exciting part is every month you will get the small series of topics covered in detail & this phase of the book covers topics like why do we use 3 x 3 kernel, why always odd kernels, why padding is needed, the importance of receptive field,why do we add layers & much more to capture the analytical thinking to build any custom neural network.

This book is for freshers, intermediate learners & for experts to master the topic more intuitively to solve real-world problems.

Note that **Easy things are hard in Vision**, vice-versa.

The book covers insight into concepts and when, how to apply them, so you can choose the right tool for the right problem. Also, it is important to learn some good examples, and let our thinking from a basic intuition to some real-time problem.

After finishing this book we can assure you that you will never say I don’t know this concept.

Remember If you can't explain the topic in layman's words then **you are just memorization not learning.**

The reason behind writing this book:

When we are learning Computer Vision went through different courses & material available but hard to find the concept why we are using padding,when to apply padding & reason behind receptive field ,while learning you can find the material with code & can built a system to detect objects but when it comes to an picture how it's is working in depth with neural network ,sad but no answer found.So,from our own research learning & building some product in Computer Vision we come up with this idea to deliver the content in unique manner for better visualization & understanding.

From our research CV engineer's are in demand but at the same time difficult to find the right one. Everyone is building projects but no one knows the picture behind the network of how it's working.This series is not only a book,it's something to build your better analytical thinking which in-turn help you out to solve different problem statements.

Remember that before writing a code ,you should understand how your network is going to learn with different edge cases then figuring out the right dataset after applying the fundamentals like kernel size,when to apply padding,when to apply max pooling,why activation function is needed & much more.

Now this is the situation when the problem begins, we are engineers let's accept it, firstly we are not interested in remembering the formula & second, no one will even ask us about that, the only thing which will help us to understand the actual logic behind the concept, the work procedure with use case where we can easily apply it to get the desired output. When we consider this need, we do not find any helping tools, & those working professionals who can really help us out, are busy with their own jobs with responsibilities.

In order to give this practical understanding to the other aspirants who really want to learn these concepts, this book is written, because in our opinions, it is not just about reading a research paper, grabbing what's new in the industry & putting in the resume to get shortlisted but it is about can we really solve a complex problem, what industry is trying to solve with the same.

How is this book different?

If we talk about previous experiences in learning, we often buy a course or a book that is promising one of these three things, “For beginners”, “with Advance Concepts”, “From beginner to advance”. But no one promises to come and understand the working with use cases in **layman's terms**, or let us discuss real-life problems & try to solve them.

Also, if we start considering some really popular books or courses, yes they are interesting and knowledgeable but either they are too fast or they are so lengthy or big that you may even leave that in-between due to lack of discipline or we just lose our interest while reading it.

Think about the most complex concept you know or let us talk about the basic CNN, just by knowing that convolution layers are for feature extraction & some matrix operations which are shown as the internal working of convolution, are we really done with this?

Well, what about considering the concepts like why we need padding? Why do we need to apply to pooling? Which type of pooling layer is good for our use case? Why always do people apply a 3x3 kernel for feature extraction? What are channels and kernels and what do they really do?

The list never ends, it was just CNN, think about the complex algorithms like object detection. We can generate lots of why, & yes we will not be able to find the answers since no one focuses on them.

The book does not contain those easy concepts which are already explained by 1000’s articles & books & not even those formulas that you can easily find in research papers. We really try to convert that hard logic of work procedure of an algorithm in layman's words so anyone can understand the logic with use cases.

One of the best part of this series is every month you will get the small handbook consists of 4 to 5 fundamental topics in layman words starting from scratch to building custom neural network while building some real time project ,in-total there will be 8 to 15 phases of this series in very unique manner which will help you gather the best knowledge.

1. Roadmap:Path for Better Understanding

If you want to master Computer Vision:"Be Single or Married:Vivek Chaudhary"

Because married people could extract more features for better learning & for single individuals they are more dedicated to finding the right feature instead of wasting their time with respect to unstable relationships(such relationships can only help you to build a bad network).

Indeed data science is giving us a lot, with the help of a huge chunk of data, we have successfully built great solutions, and we are using them now.
Considering most of the apps on your phone, you can see at least a single feature that serves you with data science.

Have we ever thought about where this need started?
Well, let us understand this with real life examples.

Consider the document scanner in our phone, most of them are able to identify the places where text exists, and while scanning and because of that feature, we save our time from cropping the image of the document.

Can we solve this problem with a rule-based system?
Here are some scenarios to think,

- Because the text is black and the paper is white, we can create a black pixel detection rule-based system and solve the problem.
- Oops, people can use different colors of text, or different ink in case of handwriting, okay fine except white, any color pixel detection system, now customers will be happy.
- No way, what about some watermarks on the page of the manufacturer, don't you think to ignore that?
- Oh my god, so just crop some area around the paper, problem solved.
- Have you considered the different background colors?
- No way, this problem is impossible to solve with rule-based systems.

Well, this discussion was seen in an engineering team of XYZ company which just launched their popular document scanner, discussing how they will solve a particular problem.

Based on the above case scenario, they came to the conclusion that solving this problem pixel-wise can not solve the entire problem. But there is a smart guy who knows a lot of image processing techniques and comes with the output.

The whole team is shocked now, and blindly started testing the solution on different cases, but here is the problem as soon as they tested slightly brighter images, or darker images, they got really poor results.

And the manager of the engineering team understands this very well, that not everyone owns a phone with a great camera.

So, can we help this team to solve this problem?

Consider any problem statement, the first step is to connect it with human, as a human how we really think about the solution, Well, as a human we know how the text looks like, does not matter whether we can recognize it or not, it means that even if we give a document written in Japanese, to a person who does not know Japanese, he will still detect the text, and at the end, the problem does not even consider the recognition.

So, how can a human detect text, even if it is not in the language he knows? The answer is simple, we have written many text documents, and seen enough examples, to understand how text really looks like.

In our understanding, the text is nothing but patterns, and even though we don't know the basic component, which is the characters of that text, we can still say it is text.

This is where the game of deep learning comes in. We take the help of neural networks to solve a problem with a lot of scenarios and rule-based systems already failed on it.

Now we need to understand it, as a human we have seen enough examples, which means all those different scenarios possible with text written on a document, different handwritings, different spaces between words in the text, different lighting conditions, and many more.

So when we consider training a neural network, we need to consider all those scenarios, and for that, we need to collect data from all those scenarios, and it is not that easy.

The problem with most of the people

When we think about neural networks, we start thinking about the algorithms and the outputs, we think the more heavy and complex the network, the better the outcomes, but do we really consider what we are feeding to this network? The data.

Well data collection is a crucial step, we need to consider the cases which are best (any system can work on these cases), the often and average once (what most of the users will be given as input, our rule-based system fails on 50% of these cases), the poor once (same as best but this time out rule-based system fails on 99% of these cases) and the worse (even the network gets failed, that 1% scenario where the network with 99% accuracy does not give results)

And after thinking about this now we need to step out and collect the data.

Once we are done with that the data cleaning comes, where we need to consider that all these cases should be balanced, which means all these different scenarios should have a similar number of examples, so our model will not get biased towards just a few scenarios because of dominance by data.

Now comes the playfield where we try to observe the data, we can play with the images, can rotate them with a certain angle, crop them, blur them, and many more, basically, as a human, we know how the car looks like, in any case, whether it is upside down or in normal position, or just 10% of the car is visible.

Model is our child we need to make sure it knows all these things before getting tested

These steps are data collection, data cleaning, and data augmentation.

Most people forget these important points to consider and directly jump to the algorithm.

Remember whatever you will need to the network, it will only give you those outputs.

Now when this algorithm part comes, we need to understand a few things.

Remember whenever we choose any algorithm, we should consider these points,

- What use cases were considered before showing the results (Basically datasets which were used, do they really meet the real-life expectations)

- What is the size of the architecture? Can we really deploy it in production? (Imagine a really heavy architecture that takes 5 seconds to process an image)

- What are the practical use cases? Do they even meet our use cases, consider a scenario where we need to make an OCR on very small fonts, and we are choosing an algorithm which was made to recognize text on banners and holding boards.

- What was the accuracy after training it, and when tested on new images, sometimes it happens we made the model, tested it on the test set and gave us 99% accuracy even in the test, but as soon as the new data comes into production, the algorithm does not give the good results.

- How much time and effort it will take basically, considering the deadline period before taking the risk.

Once these things will be done we will be successfully able to solve a particular problem.

But what about the learning part?

Well, let's talk about what we really do?

You are sitting in your college mess, suddenly your friend berry comes and says, hey billi a new type of neural network came in the market called the "the helpless networks" can learn on satellite frequencies and gives the output about the risk of crashing.

As a computer vision engineer, will you really look at it?

As a geek, it is a good step to know what is going on but imagine a plumber who calls himself a plumber but actually knows how to fix street lights.

We need to understand this, we do not need to learn everything that is available in the market.
Remember "less is more".

Consider things that your market demands, your job demands, the type of problems you solve demands.

Then pick small components and learn them precisely, learn about every technical term related to it and learn about the actual hyperparameters to tune, what about the optimization?

And at the end the amount of data required to make a scalable product.

If you know all these things, you can say proudly that you know this algorithm.
The main thing is if you are going to fight a war, you will keep all your best weapons, not a makeup kit to glow your skin.

Final Shabd

We have come across many people, who have claimed to know a lot of things, but when it comes to solving problems, they lack somewhere.

The problem is they find it really hard to connect the problem with how the human brain is solving it. It is funny to say that a lot of us being a human don't know how a human brain works, but it is true.

Consider a case of detecting helmet, let's take two approaches

Approach 1

- Downloading the data set from available resources like Kaggle
- Finding the best object detection algorithm in the market (took nearly 4 days)
- Testing the model
- Results are not satisfying
- Breaking the PC :(

Approach 2

- Asking the client where they will deploy the model
- Taking the information of existing system and reasons to switch to a new model
- Consider all the edge cases like different colors of helmets, different types of helmets, different vehicles where people wear helmets, edge cases like covering the face with a cloth, covering the head with a cloth, and many more.
- Collecting the real-time data and balancing it.
- Now finding a proper algorithm
- Testing successful
- Labeled as a good computer vision engineer

You need to decide which guy you are, the first one or the second one.

2.The Convolution Procedure

Snake1- Bro this is an apple (FUSS FUSS)
Snake2- Okay but can you give me any proof? (FUSS FUSS FUSS)
Snake1 - What do you mean, mamma snake told us (FUSS FUSS)
Snake2- 🐍🐍🐍🐍🐍🐍

These two poor snakes fought with each other, but have you ever considered we humans often do this?

Think once?

Let's compare human vision with computer vision for a second & think about it peacefully.

Source

Did you remember in our childhood we have been taught while pointing out the fruits & at the same time visualizing something like 'A' for apple,'B' for the ball so on….if you see this our eyes get convolved with different objects with visualization & storing that information that how apple looks like in our brain known to be as a convolution in simple words.

Same as if you don't see the object which means you are not convolving through to identify or classify the surroundings.

Did you sometimes think that as human beings we don't give that much importance to the colors of an object at some point instead of focusing on the shape, size & texture of the object to understand how it looks like that is why if you see the apple of green color , red color,yellow-reddish color you can still understand that it's an apple because you can differentiate between shape, size, the texture of different objects like apple, mango, etc.

& what if we focus only on colors which could have fed wrong information like apples are always red, etc….are you getting it!!!! but it doesn't matter that color doesn't have

importance at all but we focus on shape, size & texture instead at first to understand much better because the same object could be of different colors which may confuse our brain at some point.

If you are looking to buy a refined oil of fortune company then how do you behave:we already have the information stored in our brain about package of fortune looks like,will you pick every refined oil package,then read the name after get to know ohhh man it's not fortune one(we as human being doesn't behave like this) instead here our eyes are focused on extracting the item looking for(fortune refined oil) ,we already have the stored information in our brain,so:in simple we jump instantly from one rack to another to find the specific product looking for & in simple you are convolving through different items based on the specific oil with the help of already stored information & extracting the right one information instantly hence know to be as convolution(one of the important aspect of human life).

Now let us take this to the next level,

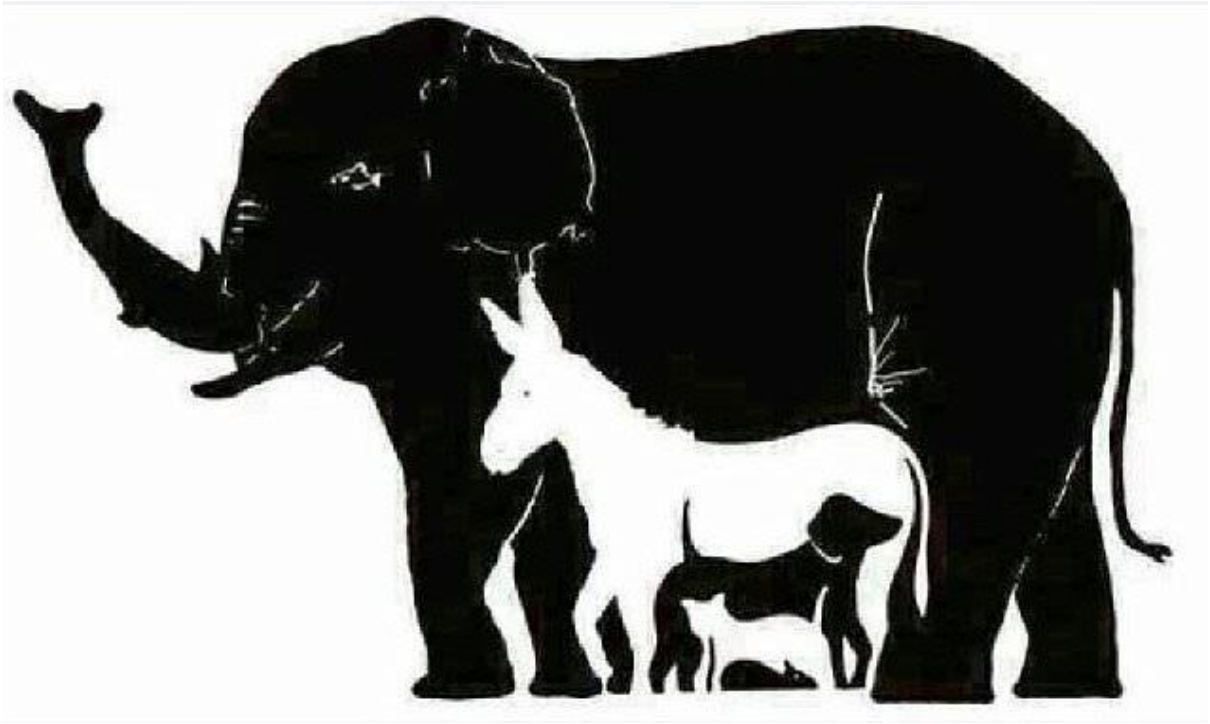


Image-source

Consider the above image, you can see an elephant, a dog, a cat, and a donkey without any issue, consider it like normal human vision, but what if I tell you there are 15+ animals in this image, confused right?

Well now to figure it out, instead of solving this riddle, try to understand the process that your brain is following right now, let's analyze it.

On the first hand we are trying to see the edges of these animals in case they are building any other animal or not, for example, if you analyze the trunk(nose) of elephants edges are building a fish, but did you notice it on the first hand? No right

Now analyze more you will see all these animals in the below image

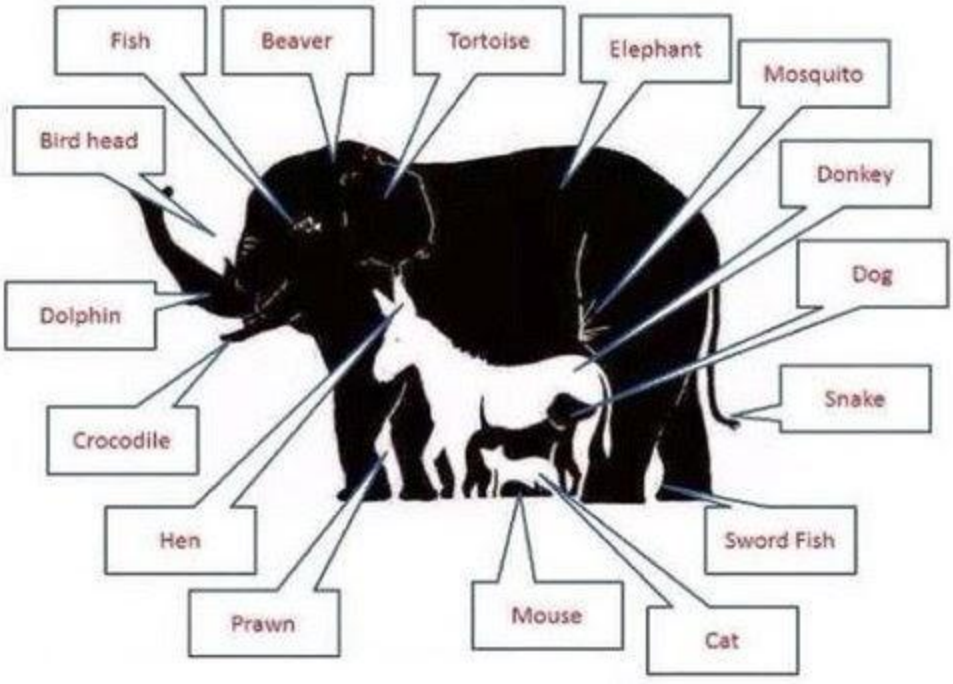


Image-source

Well, till now we were trying to connect the dots between human vision and computer vision, but put computer vision aside, and think about human vision, didn't we apply a convolution layer first which was extracting edges of animals in order to build the shape same as we do in a convolution neural network.

Assume you are entering a room which you have never seen before, will you directly jump on the place you want to go or you will analyze the things first, the small objects available in the room & then reach the place you want to go.

Another example can be when you cross a busy road full of traffic, don't you first notice each vehicle.

Well since convolution neural networks learn from data, they follow the same step, first, they try to analyze the smallest component, the edges, and understand the basic build of objects, as we know starting convolution layers extract the edges and gradients first, later the patterns, parts of objects and finally the full object comes in the picture.

Since now we have a clear understanding of the procedure of vision in our hands, let us talk about something interesting.

"Convolution operation importance is same as the water for a human being"

In childhood when a child is just a few months old, they never visualize the objects like how we do. A small child can only see a few edges and patterns of surroundings,this is one of the reasons why kids cry a lot when they are babies.

They see blurry shapes of different objects & even science says they only have eyesight in the range of 20/200 to 20/400.
Do you think about what is going on there, well they are convolving things, you can consider that early age as starting convolution layers.

2.Channels & Kernels:The Sidepath

What comes into our heads when we think about image classification?
Convolutional Neural Networks right?

But have we ever focused on the components? Now do not give that look, we know convolution layers, max-pooling, normalization, dropout….. Even what individuals follow blindly build the networks without the foundation behind the neural network & that is what we are going to cover the basic building block of networks with intuition.

But it is always better to understand the roots.

When we talk about images, we talk about information.

What is information?

Well, information tells about the content available in the image.

For example, If you consider the above image, you can see the handsome hunk Raju standing in a stylish pose, wearing really expensive sunglasses, really confident, the man every girl dreams for.

Now let us see from the perspective of neural networks,

Networks don't see it as information, in the words of deep learning we call it features.

A system will tell, a person is standing in the image, having a pose (pose estimation), looking confident (emotion detection), wearing glasses, and containing things in the background (object detection).

Can we figure out the difference?

Well, where we humans connect small dots and create a story, computers focus in a more technical way and try to see components not directly start singing the story.

When computers see an image, they don't directly jump to a conclusion that an object is there in the image.

They try to see small edges and gradients first, then the textures and patterns. Based on those observations, they try to build parts of objects, and finally, these parts are combined to form an object.

If you consider this whole procedure, you need someone who can first see these edges, gradients, textures, patterns, and parts of the object first hand in a neural network and later save all this so we can process it.

This is where the concept of kernels and channels comes into the picture.

Kernels & channels

A kernel is a feature extractor that extracts features in the image, it will extract the important features like the tail and ear of a dog and remove the noise like background and channels will store this extracted information which will act as an input for the next kernel (or layer).

As we have different features, we keep increasing the channels so that we can keep these features in separate containers like a separate channel for the ear of a german shepherd (erect ears) and another for the ear of a beagle (flaccid ears).

Please do not get confused about channels and the number of channels, a channel is acting as a container, containing the information, a number of channels define how many channels we have for storing the extracted information by the kernel.

Let us take an example and understand this whole concept more clearly.
Let us say we go to a restaurant, where we see the menu to order something for us.

When we pick the menu we can see two things, the veg & non-veg sections.

Obviously, veg contains only vegetarian dishes and non-veg contains only non-vegetarian dishes.

In the above case, we are behaving exactly like a kernel where you are extracting that information from the menu, and pages that contain the veg and non-veg dish information are two channels keeping similar information.

Let's say we decided to eat veg, in this case, we are acting as a kernel and going in the veg channel, trying to extract more information about the menu and dishes. We see there are four main things, first the tandoor, second curry, third rice, and fourth chapati, tandoor contains many items like paneer tandoori, etc…, curry contains potato curry, mix veg, rice contains fried rice, and biryani and so on.

If you consider this divided menu all these things are features and which are contained by the veg channel, so you act as a kernel and try to extract similar information again.

You can play this game the whole day.

Now let's understand in a more informative way

In Simple words,Take an example of Emotion Detection & here if we say what are the important Features for this particular problem statement as follows:

1)Difference between two eyebrows is a feature,

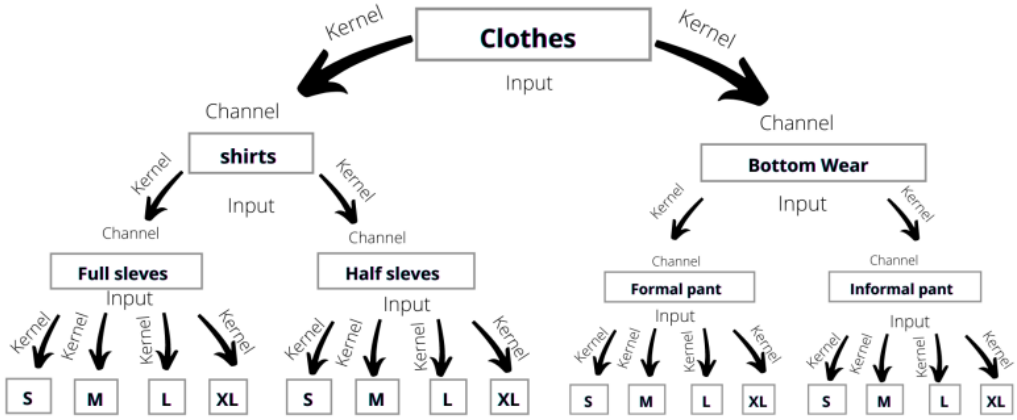
2.)width of a eyebrow is a feature &

3.)Shape of the lip is also a feature.

And what the kernel will do here is, it will extract the features from the particular image for emotion detection like width of an eyebrow & shape of a lip etc.

An output of a kernel will be stored in a channel.

Let's Understand more with some real world example in layman words from the below image:



Let's Understand Above Image in Detail with respect to Channel,Kernel & Feature:

1.)As seen in the image we have given an input image to our network which will classify it into shirts & bottom wear.

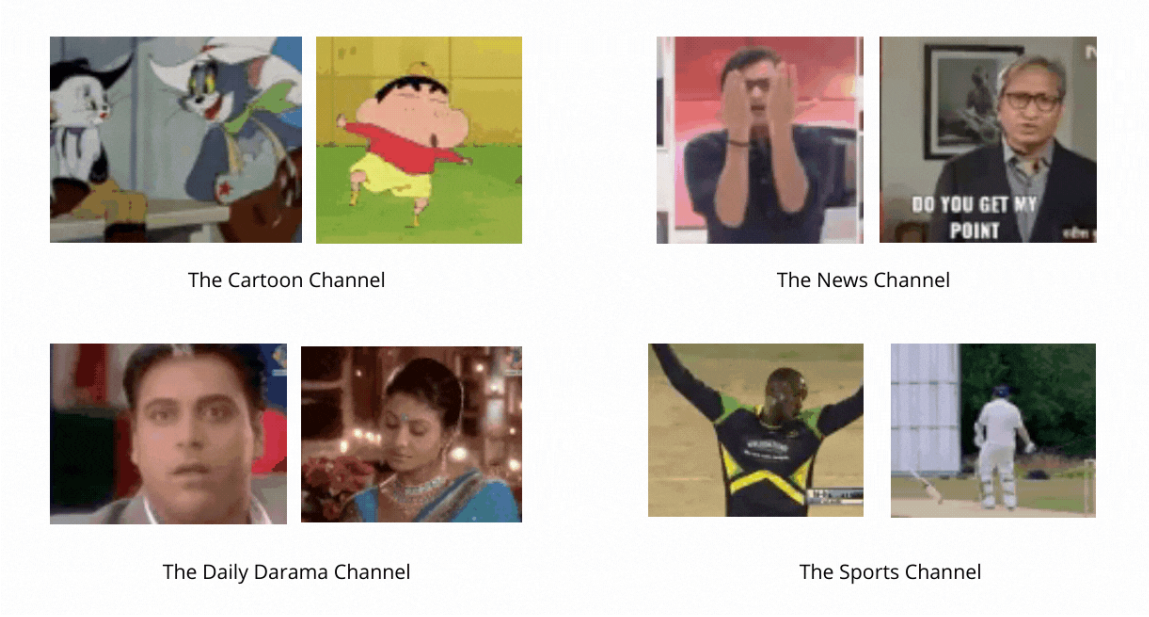
2.)Now as you can see kernel will extract features like full shelves , half shelves,formal pant & informal pant

3.)Features extracted in point 2 by kernel will be stored in the channel as you can observe full shelves & half shelves are in shirts channel,formal pants & informal pants are under bottom wear channel.

4.)Every channel is the input for the next layer(in DNN) if you can observe closely.

You can even consider the TV channels where some are dedicated to cartoons, some for news, some for daily drama shows, and more.

Similarly, in a CNN the channels in initial layers will keep general information like we can have three channels in the first layers for keeping veg, non-veg, and beverages, now we add more channels and divide the veg channel to say dry sabzi and sabzi with gravy, for non-veg we can have channels for mutton and chicken dishes and for cold and hot beverages and so on. **Now, we can relate how we keep separate information as we move down in a network. Initially, the channels contain general information and they become more specific towards a particular class as we move down in the network.**

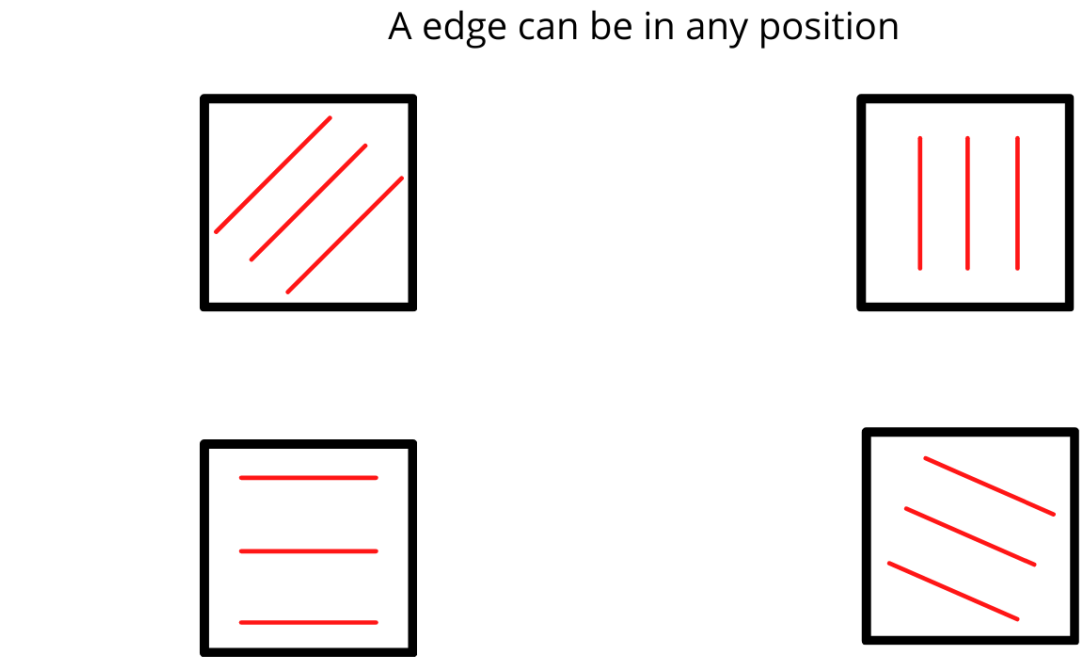


The Technical Side

Basically, when we consider these cases on images, we see the edge is just not an edge, it can have different shapes.

It can be straight, tilted, horizontal, and many more.
Also, you have gradients at the same time, when kernels store all these they store them in separate channels.

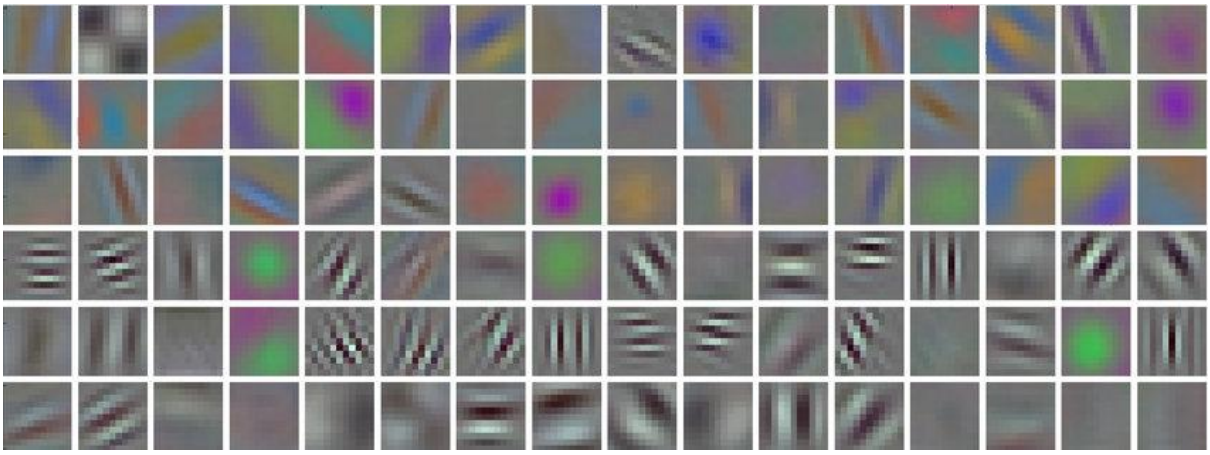
A channel can be only dedicated to straight edges, another is just containing horizontal, and another one is just containing tilted edges.



At the same time, some channels contain gradients and color shades.

Then there will be dedicated channels for storing the patterns in the image, for example, some matrix kind of structure on a particular object, for example, checks on a man's shirt.

We can consider this image for a better understanding



Source

In the above image, we can see similar edges are stored in one channel at the same time, color gradients are holding separate channels, and in some channels, we can even notice patterns.

With all these conclusions we later take all these channels and send them to the next layer of the neural network and the process goes on.

Thank You