Session 1 - Fundamentals of Artificial Intelligence

- Due Jan 27 by 9am
- Points 0
- Available after Jan 20 at 12pm

Session 1 - Fundamentals of AI

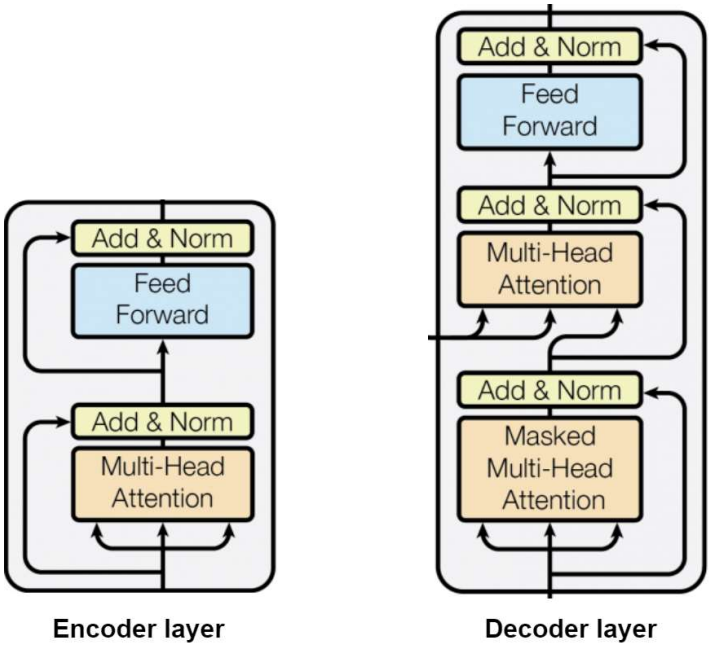
OR

BREAK IT DOWN!!

Today we'll learn more about the ERA course, and its structure and get an introduction to neural network concepts, data representation for images, text, and audio, conversion of spatial to temporal data and vice-versa, a little about convolutions, and fully connected layers, and finally forward propagation.

ERA V2 and why we needed to merge EVA and END

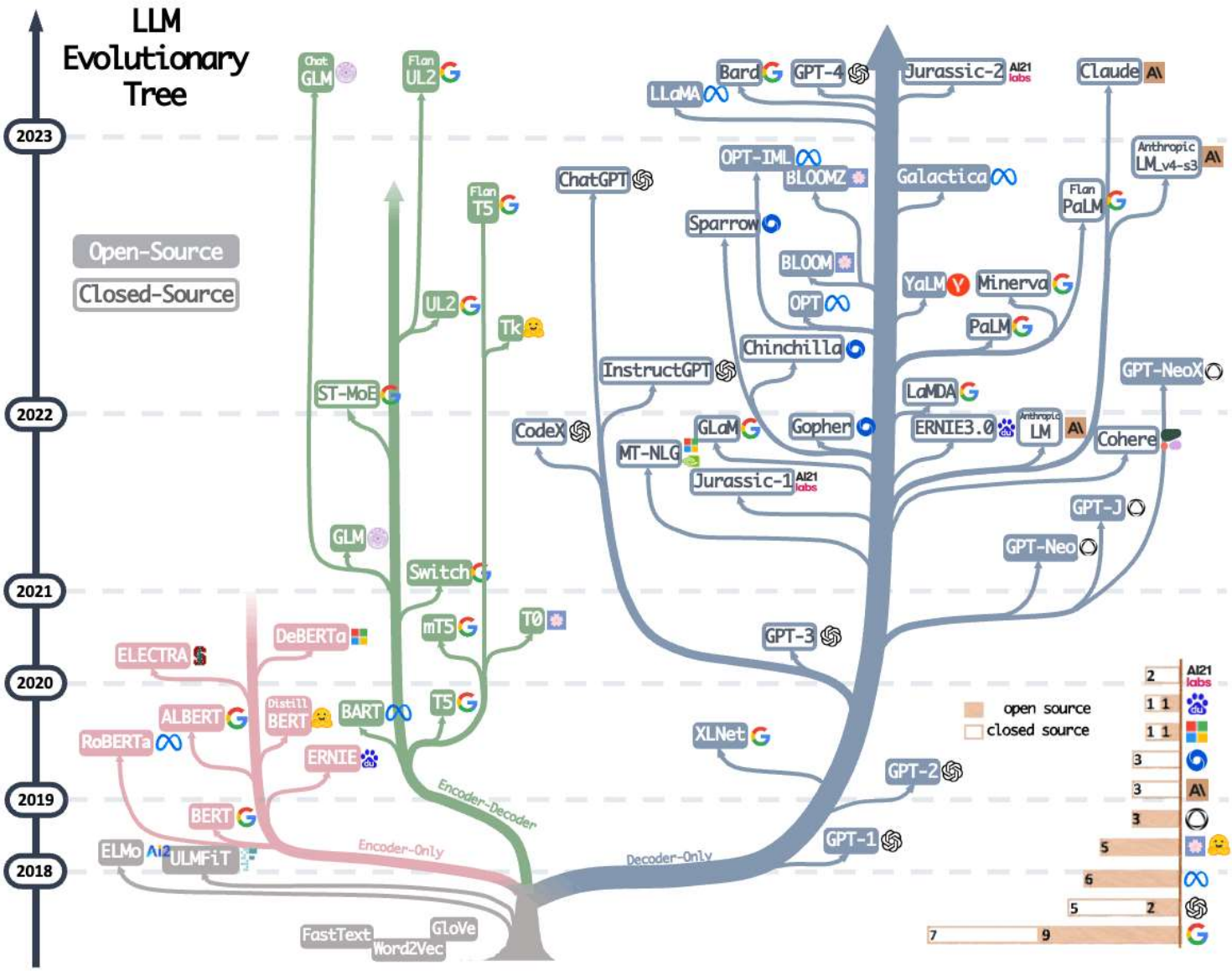
Since 2017, transformer-based architectures have been revolutionizing the foundations of Vision AI, which was based on convolutions, and NLP, which relied on RNN/LSTM architectures.



Transformers

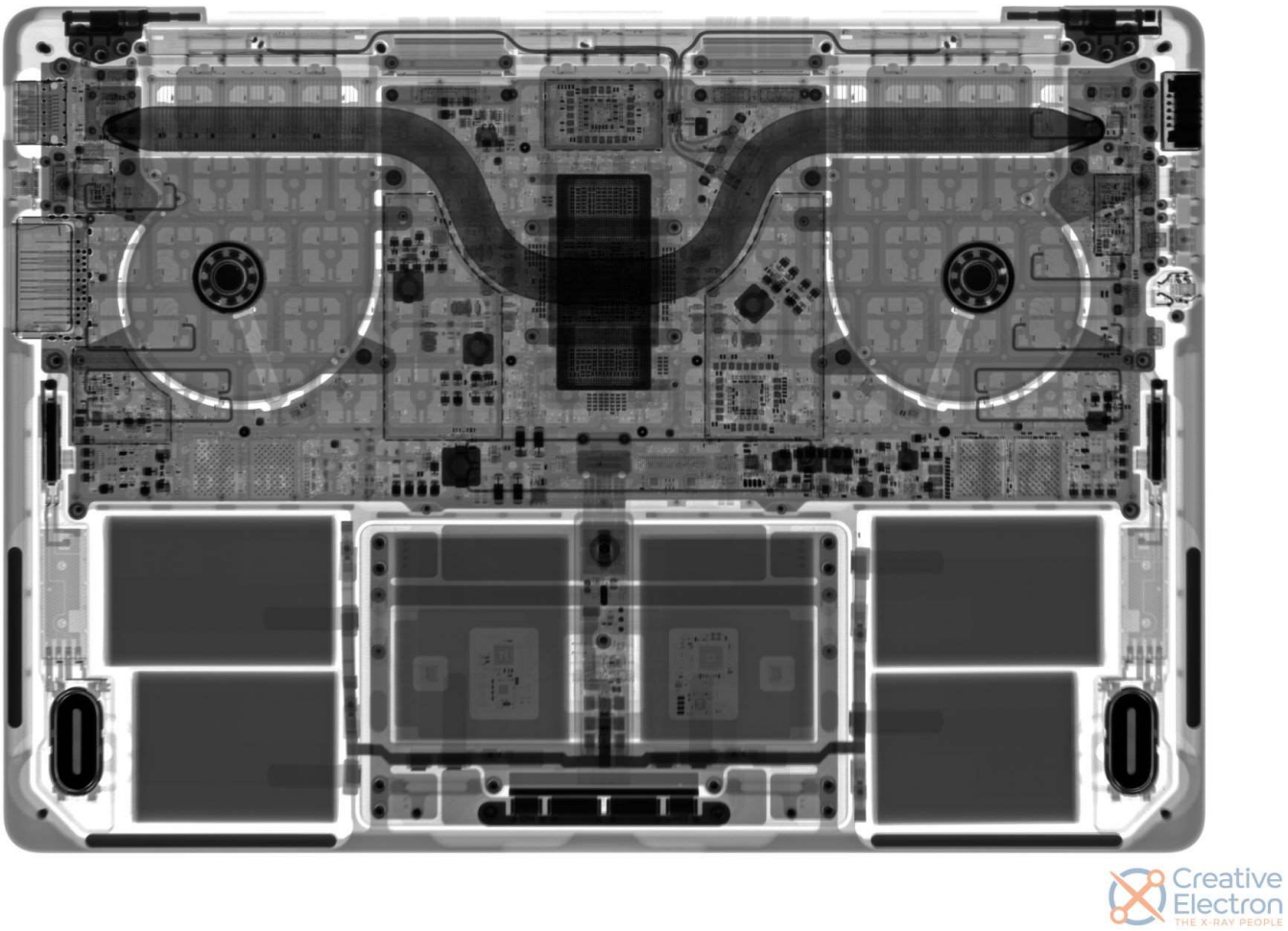
However, separate transformers were utilized for vision, images, and other domains. Over the past 24 months, there has been a significant push towards creating a single

architecture or model for multiple domains, known as multi-modal architectures.



Source (https://github.com/Mooler0410/LLMsPracticalGuide)

These models offer more than just the ability to solve multiple problems simultaneously; when trained on multi-modal data, they tend to solve all problems more effectively. The new state-of-the-art results being set are not just marginally better, but significantly better!

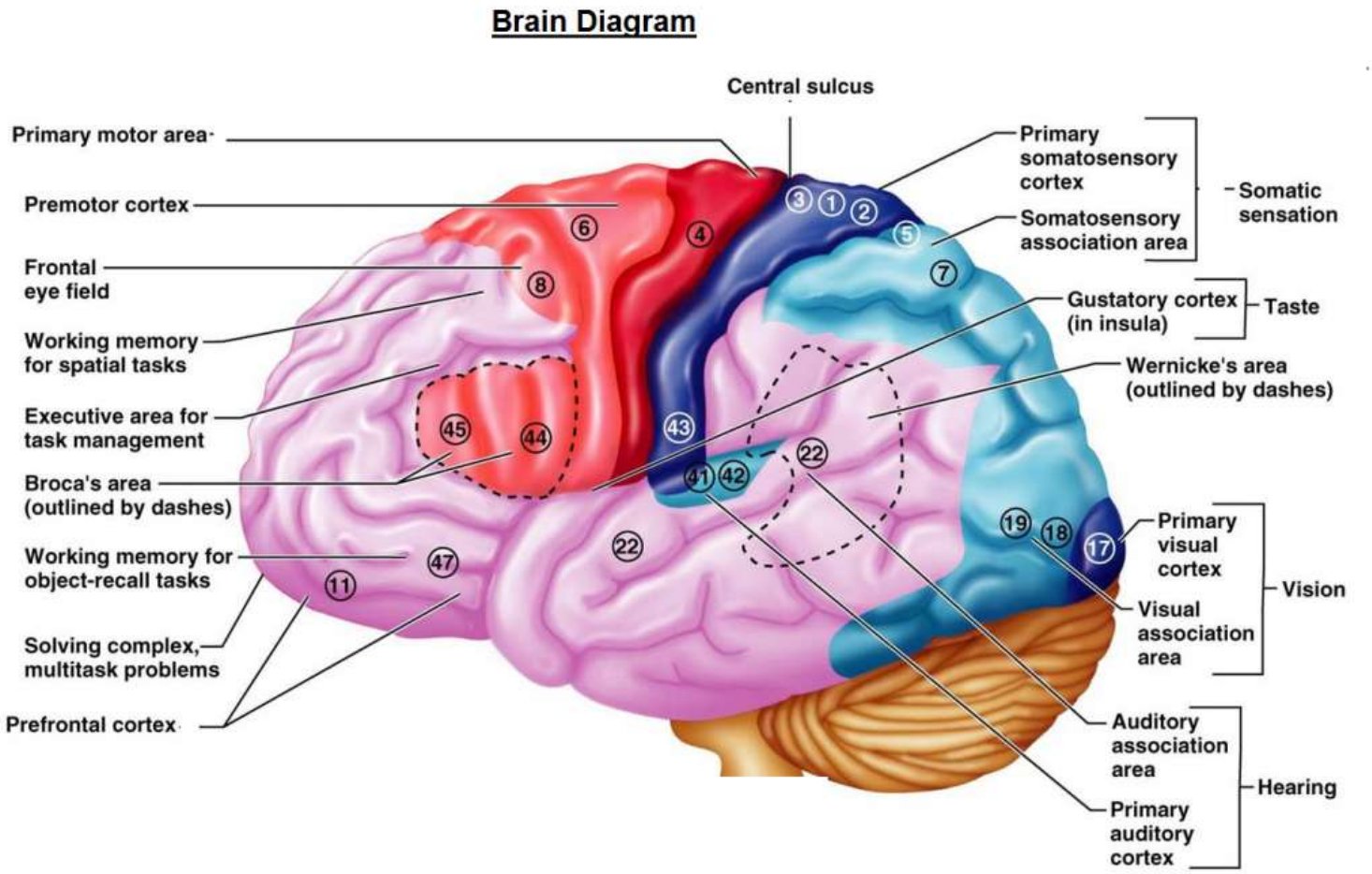


Consider this analogy: a picture is worth a thousand words! There is a vast amount of data within a single image that could be used to train an NLP model. Instead of writing about it, why not provide the model with the image and let it figure things out?

Macbook Page on Wikipedia

Similarly, there are numerous classes and types of interactions between objects that cannot be discerned solely from images. Why not share as much text about all the images as possible and let the model identify these classes, subclasses, and interactions?

Drawing from a similar concept, the human brain possesses a single architecture capable of handling different kinds of data in distinct regions, with substantial interaction between these areas.



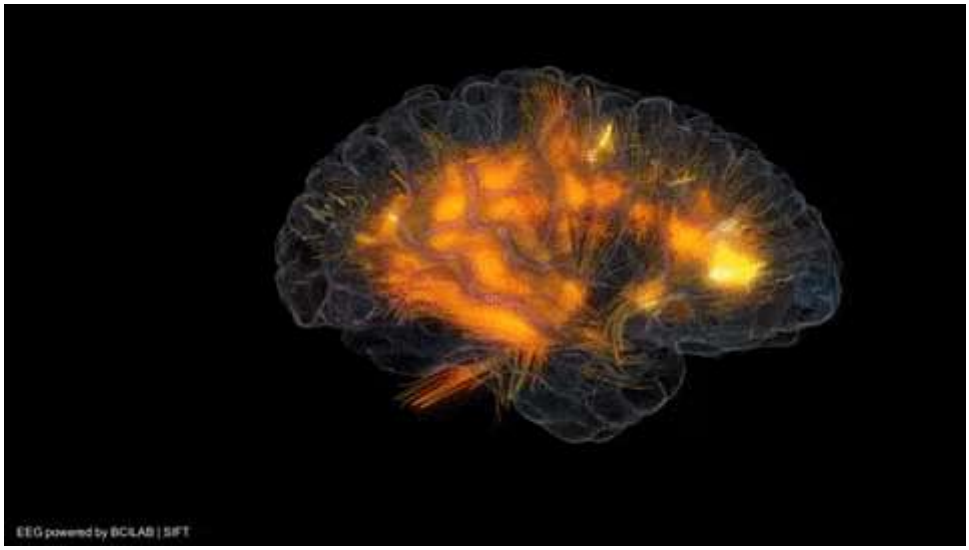
Anatomy of procession regions in our brain.

After 8 EVA versions and 3 END versions, where we constantly updated the course, it was clear, that the fields have merged, and hence a new ERA is born!

been using this image for years to share with students what exactly is their target after finishing this course:

This is how I want you to feel after you read a new paper or code about new architecture after finishing the course!

Building the Intuition: Data Representation



*An ultra-dense connected network of flowing information **inwards!***

What do we observe in the image above?

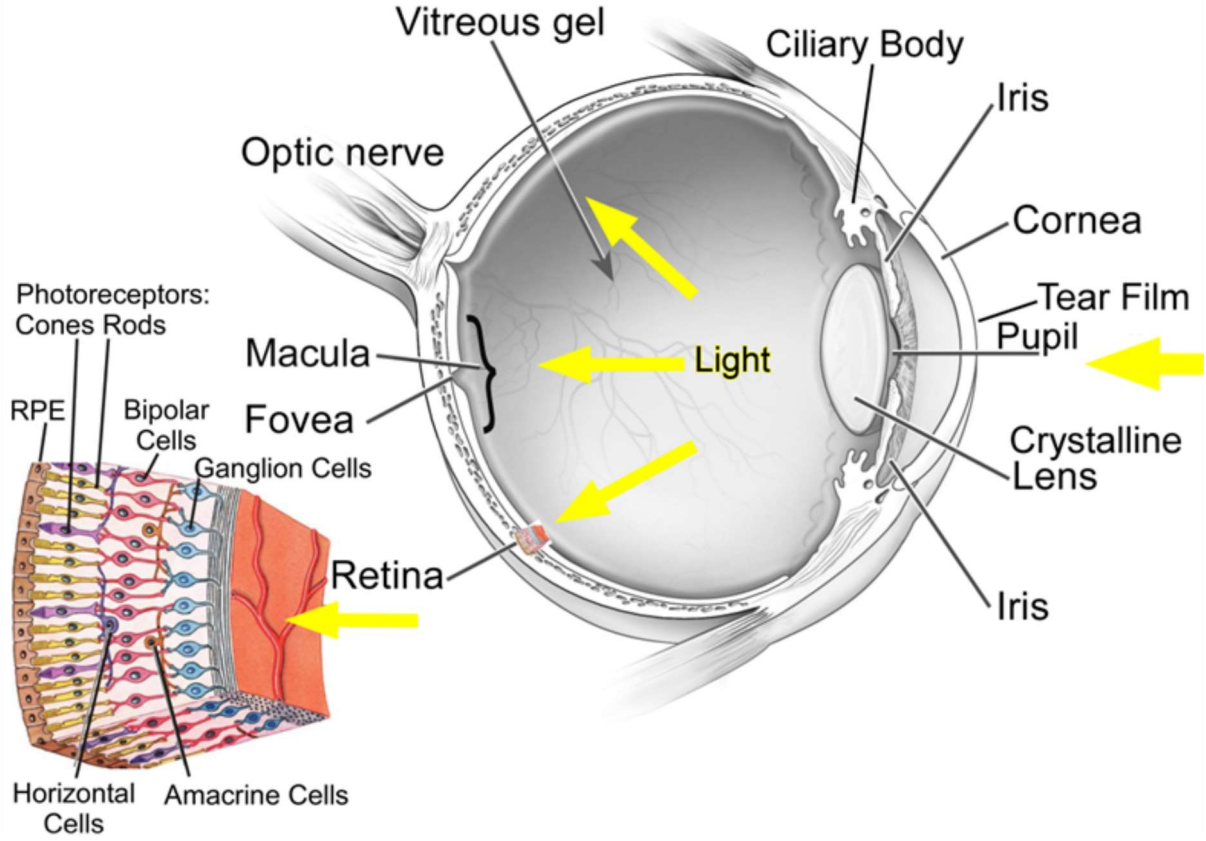
In this course, we will focus on vision and text (with a bit of audio as well). Let us see how vision is represented in human brains first.

Human Eye

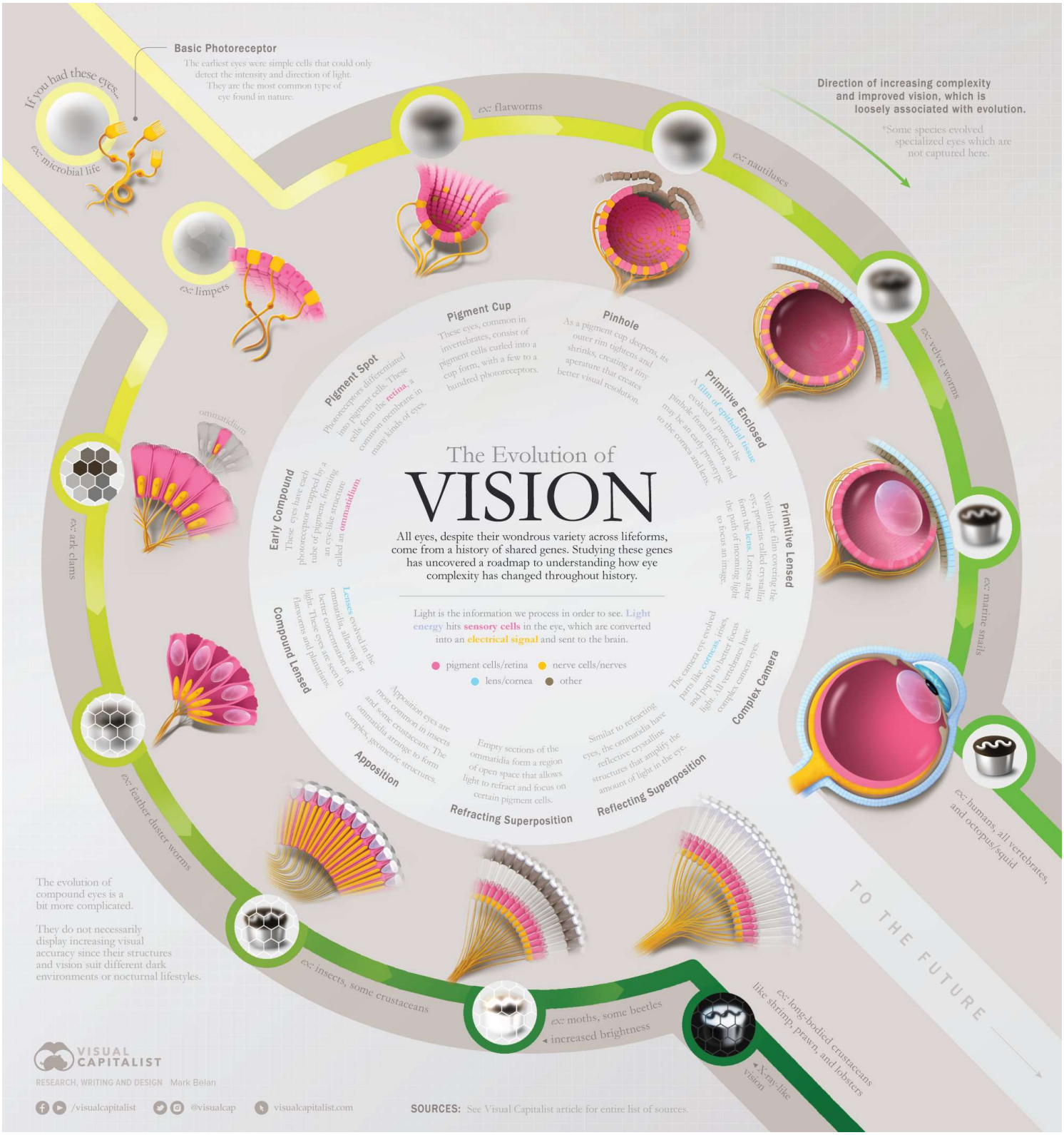
Human Eye Made on MidJourney!

The human eye is the result of billions of years of evolution **(read this amazing post on the evolution of eyes) (https://www.visualcapitalist.com/eye-evolution/)** , and what our eyes and brain do is just magical!

Those beautiful orange structures are actually muscles that pull the lens to help us focus, exactly like, how you'd use a DSLR Camera lens. Light falls on the photo sensors in the center (just as it would on a DSLR/phone)

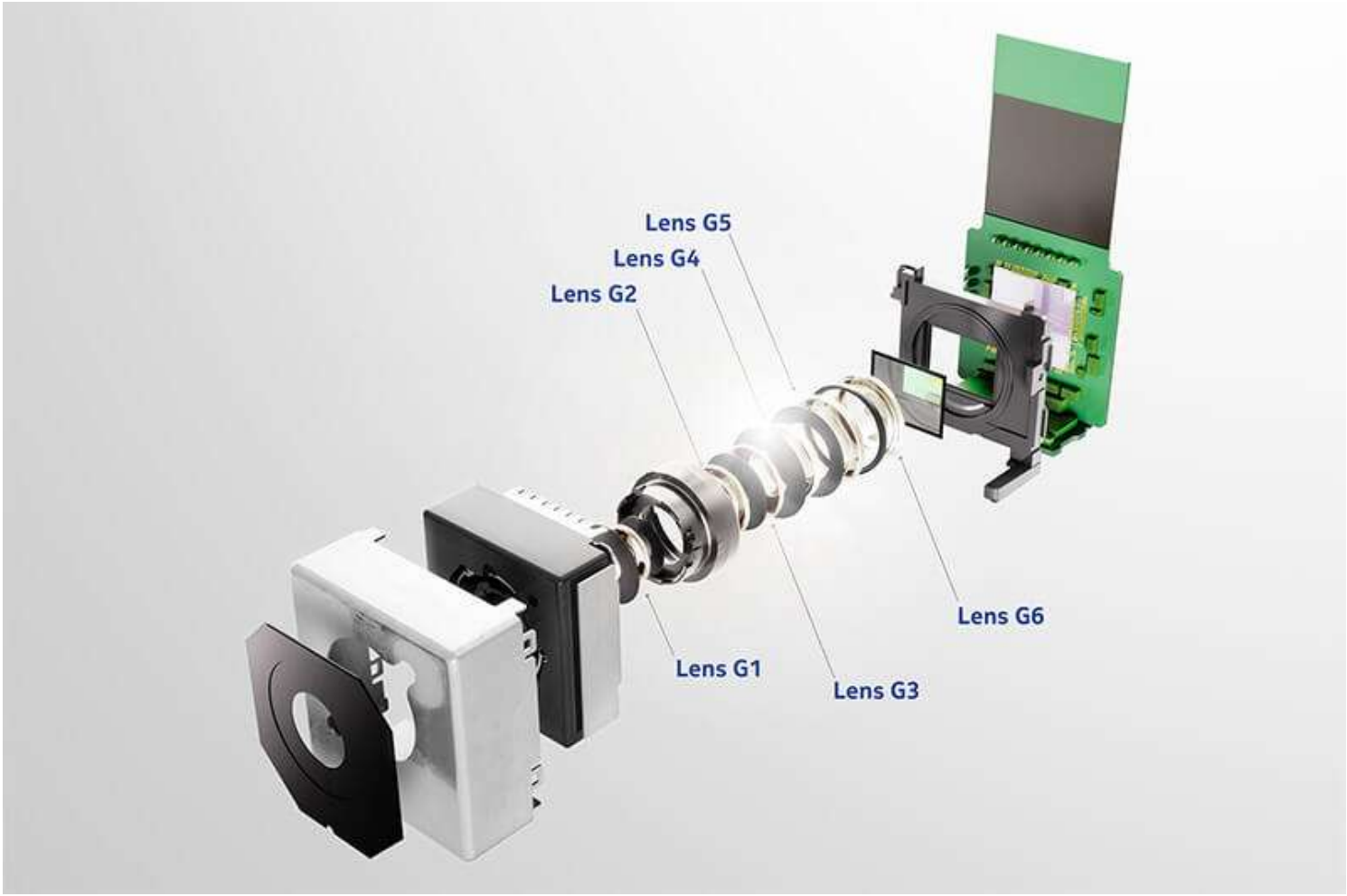


Source (https://www.intechopen.com/chapters/26714)



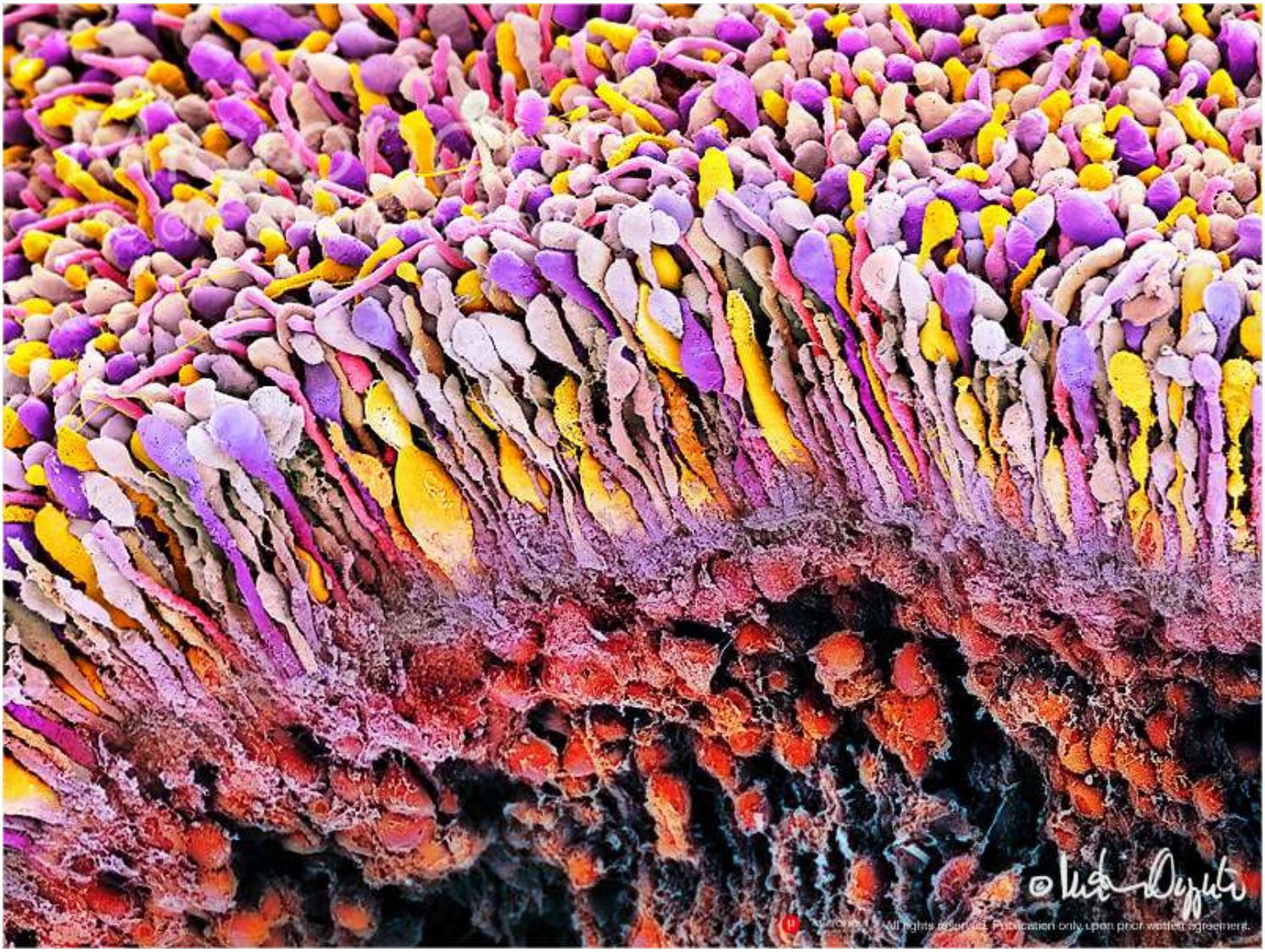
Source (https://www.visualcapitalist.com/eye-evolution/)

Similar to our eyes, humans have invented the whole sensing system and this is what it looks like:



A Camera lens and sensor system

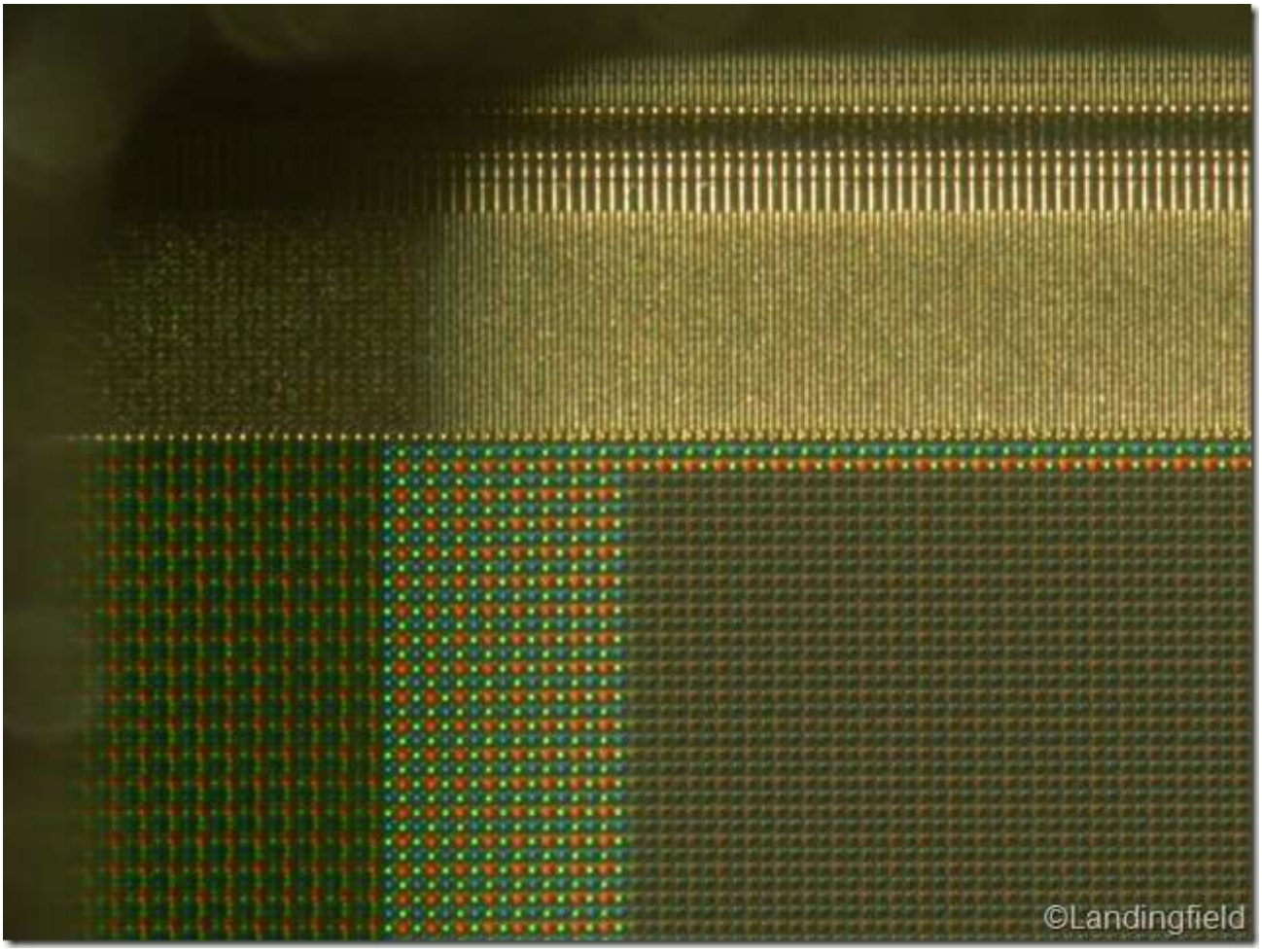
If we use an electron microscope and look at our retina, this is what we'll get:



Rods and Cones under electron microscope

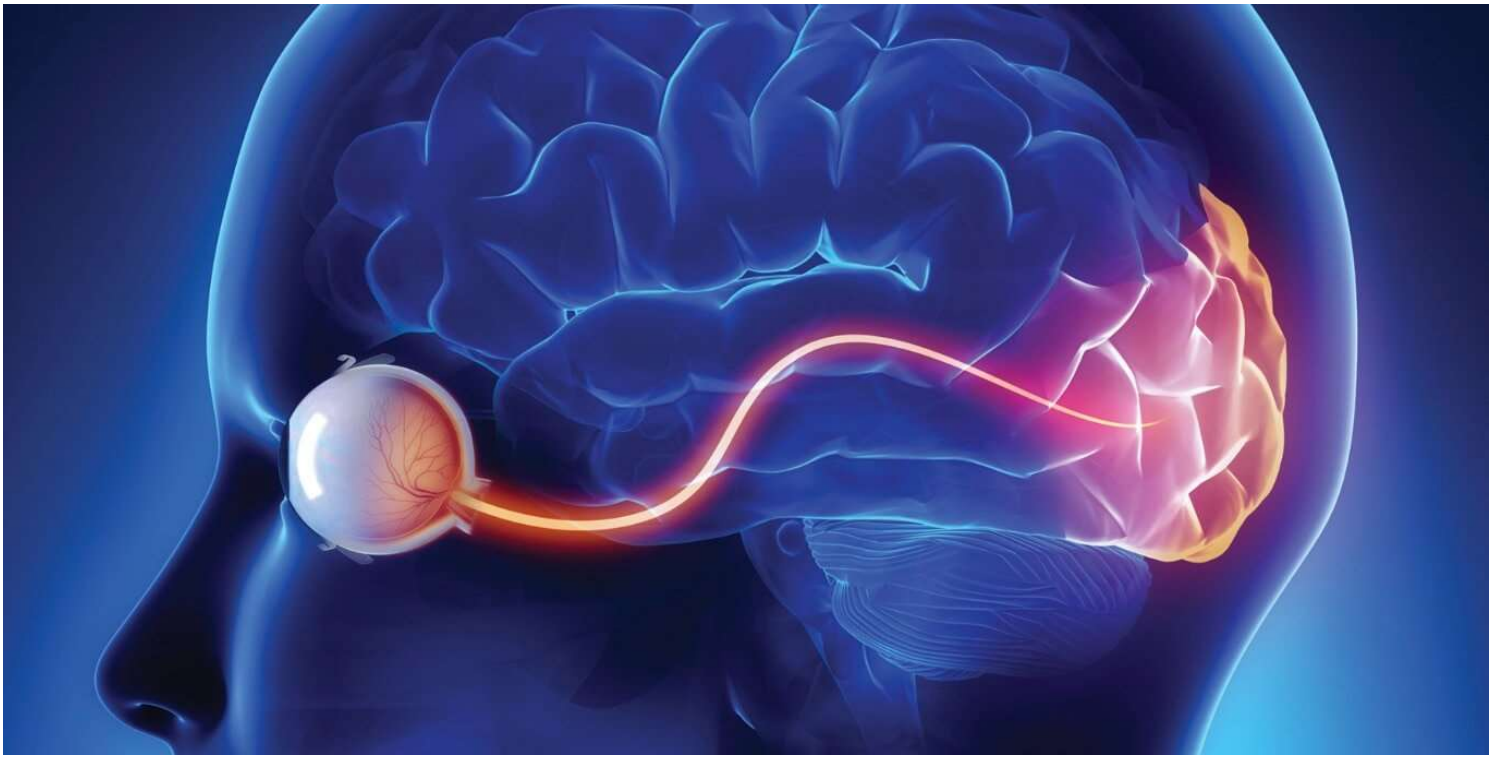
Rods: Cones ratio is 20:1. Rods see BnW and Cones see colors.

This is what a camera sensor looks like:



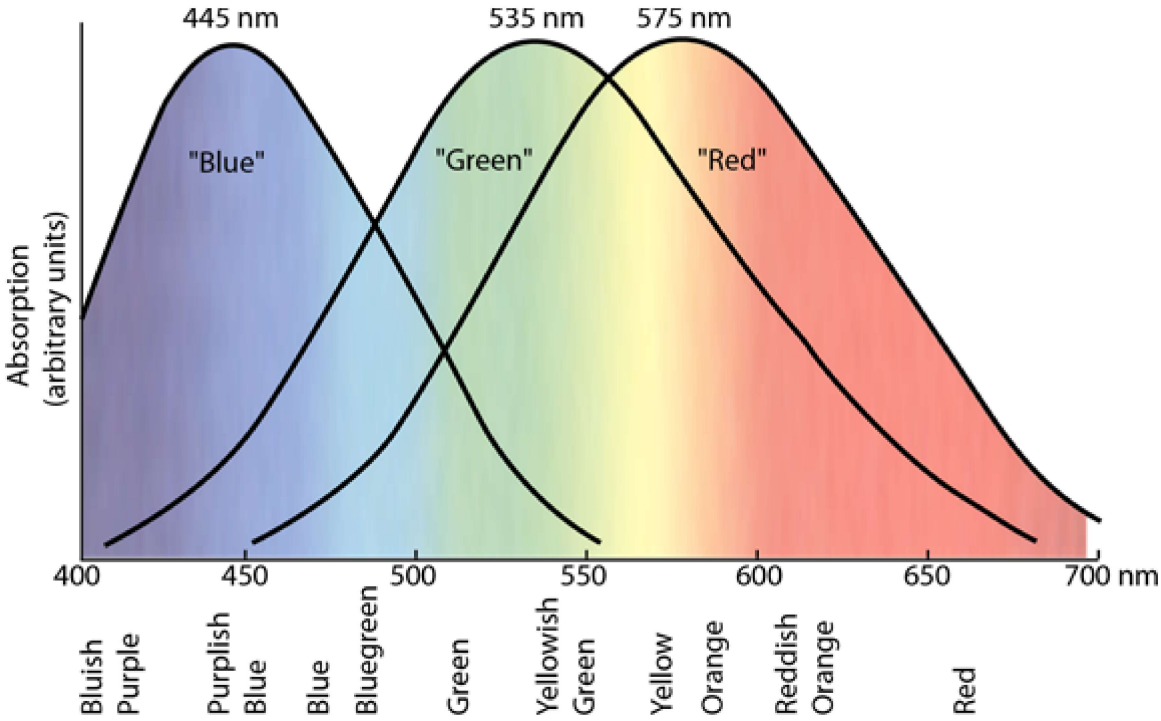
Camera CMOS sensor under a microscope

And just like a camera, the human eye also has **three types (https://askabiologist.asu.edu/rods-and-cones#:~:text=We%20have%20three%20types%20of,blue%2C%20green%2C%20and%20red)** of cones: red, green, and blue! They just need a lot of light to work, and hence so many rods. Our optic nerve is connected to our visual cortex, where the processing takes place:



The visual path from the retina to the visual cortex

One thing that we should know is, that saying that our cones see red, green, and blue, is misleading. They are sensitive to these colors, and many around them, as can be seen in this image:

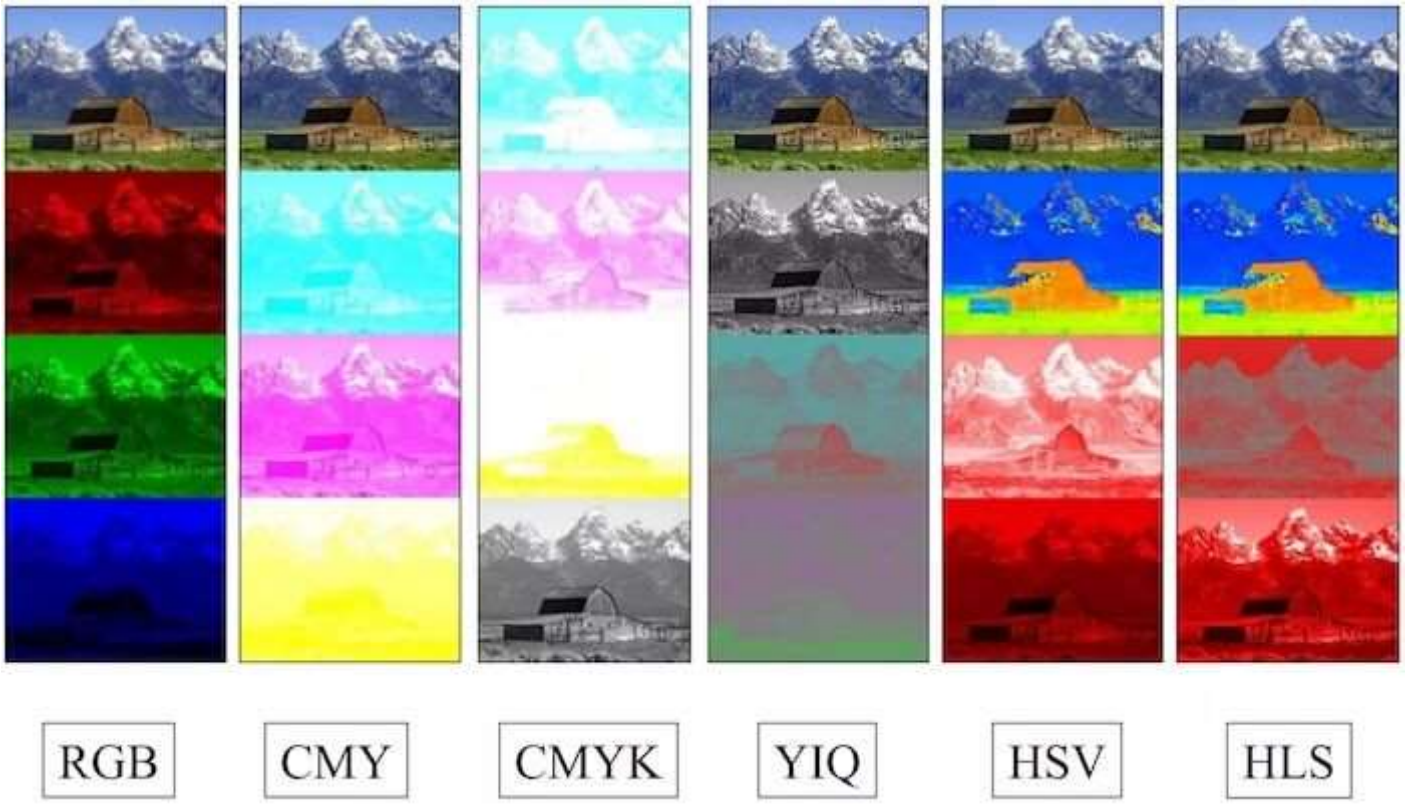


Source (http://hyperphysics.phy-astr.gsu.edu/hbase/vision/colcon.html)

Understanding this concept is super critical for us!

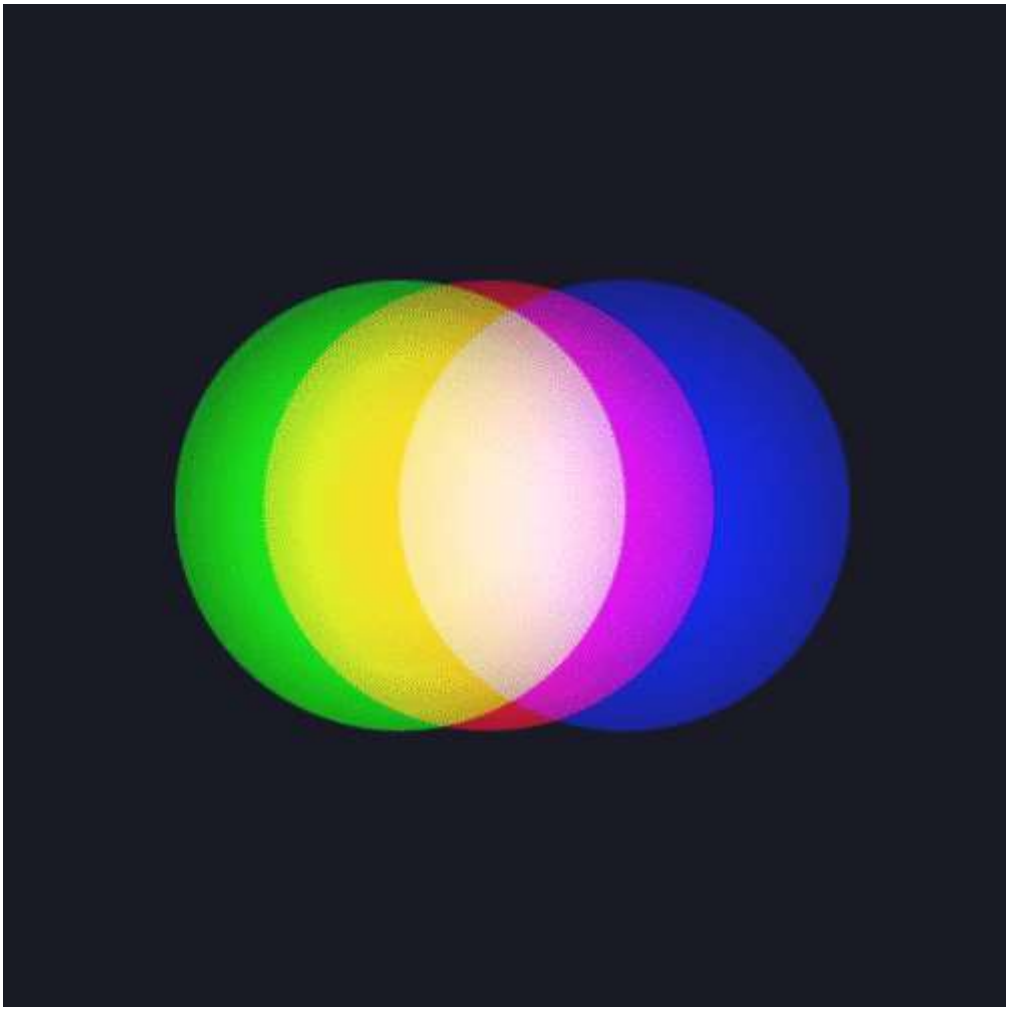
Channels or Breaking it down!

Any color can be made through a combination of different kinds of "primary color" combinations. RGB is not the only model. We can use CMYK, CMY, YIQ, LAB, HSV, HSL, HVC, Munsell, YUV, YCbCr, HSI, CIE, XYZ, etc!



Representing same image in different color models.

What I want you to focus on, is the fact that there are numerous ways of representing the same complex data.



RGB creates different colors

And you must realize this.

On a side note:

stujp × स्टजप

I want you to realize the fact that nearly the same **word sound** can be made in 293 different writing systems! And the same idea can be represented in any of the 19500 languages and dialects, just in **India** (https://www.thehindubusinessline.com/news/variety/india-is-home-to-more-than-19500-mother-tongues/article24305725.ece) !

Again the emphasis here is to understand the fact, that a "complex" concept could be represented in many many forms.

Just like different units of measurement:

For distance

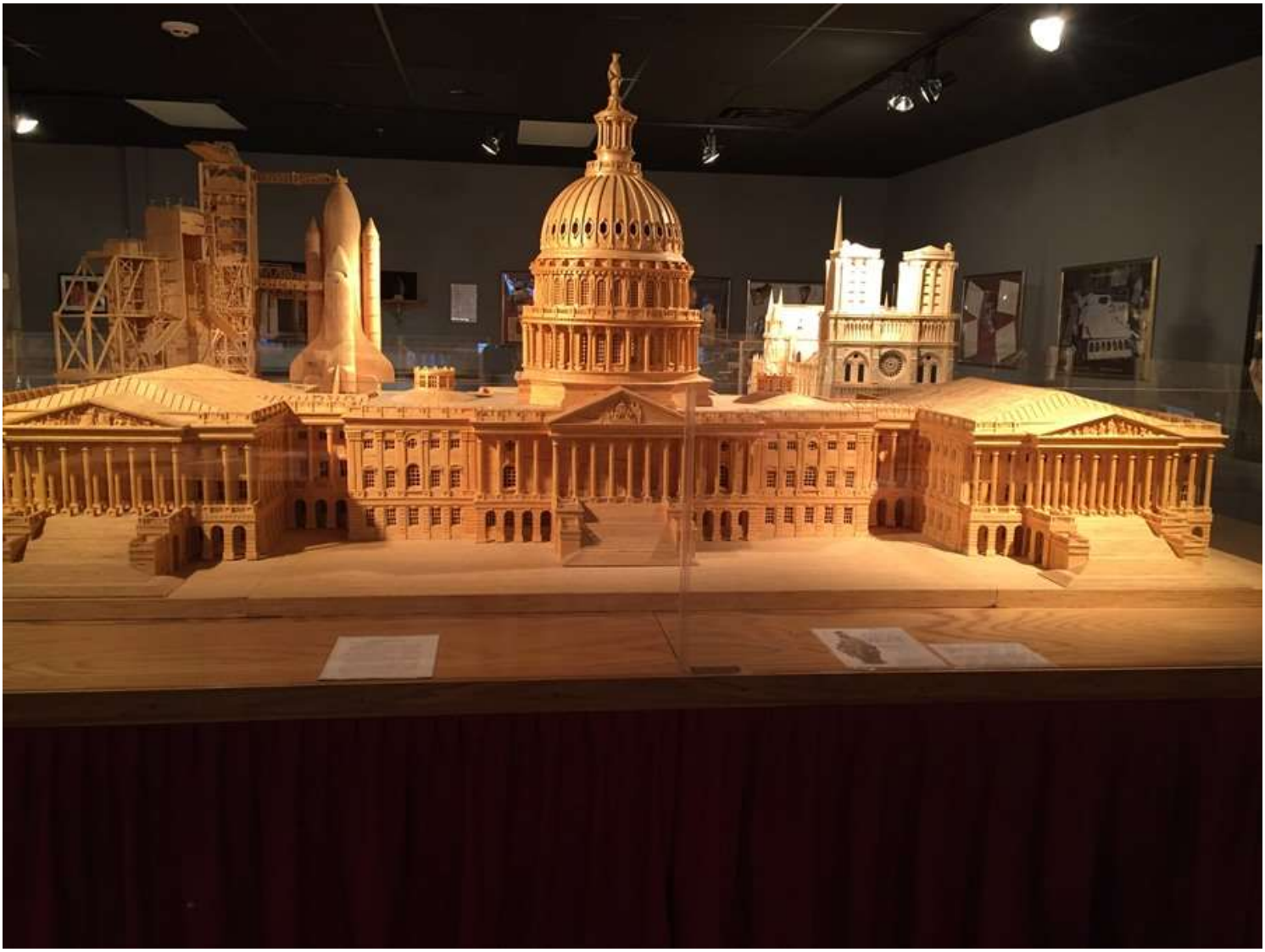
Kilometer (km), Meter (m), Centimeter (cm), Millimeter (mm), Mile (mi), Yard (yd), Foot (ft), Inch (in), Nautical mile (nm), Astronomical Unit (AU), Light-year (ly)

For Weight

Kilogram (kg), Gram (g), Milligram (mg), Microgram (µg), Pound (lb), Ounce (oz), Carat (ct), Ton (t), Metric tonne (t), Atomic mass unit (amu), Moles (M)

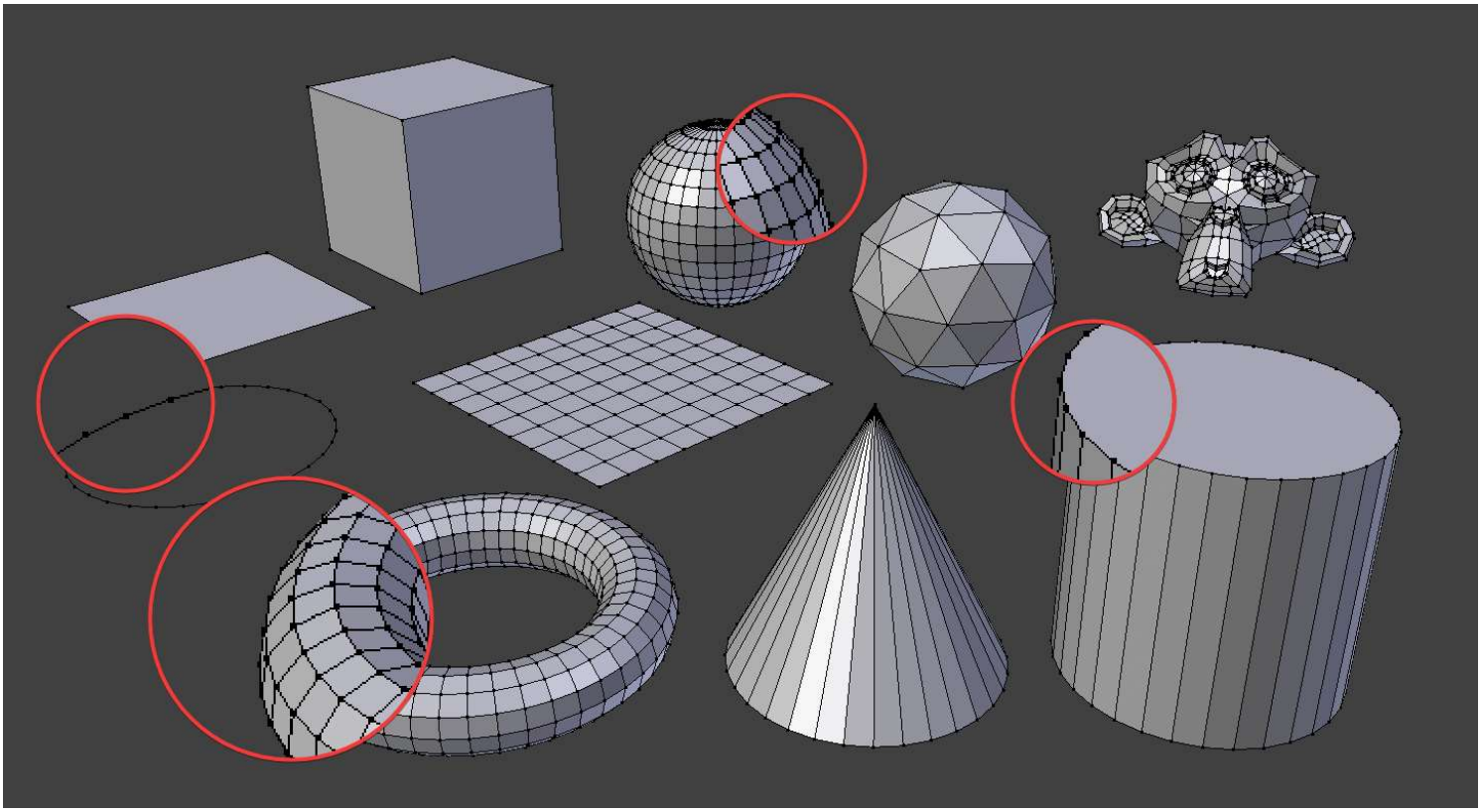
Expressing the same concept in these different units does not change the underlying value, though the context in which we might use them is different.

Now I want you to guess what is this



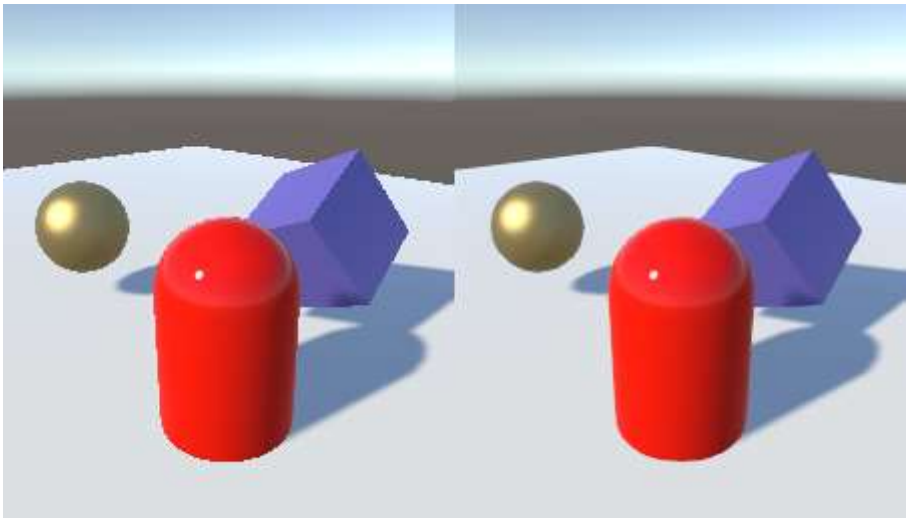
Source **(https://www.matchstickmarvels.com/the-models/)**

In 3D graphics, we actually never see curved edges. Ultimately it is broken down in a quantized line!



Everything in graphics in creating using a line.

We have to use a lot of tricks to fake smoothness



LeftL: Original image. Right: Antialiased image

So we have seen a few simpler "primary" representations (like colors, edges, and characters) for complex things (like images, and concepts).

And there is an ocean of different kinds of representations between these primary and complex representations.

For instance

Edges & Gradients, Textures, and Patterns, Part of Objects and Objects

Characters, Words, Sentences, Paragraphs, Book

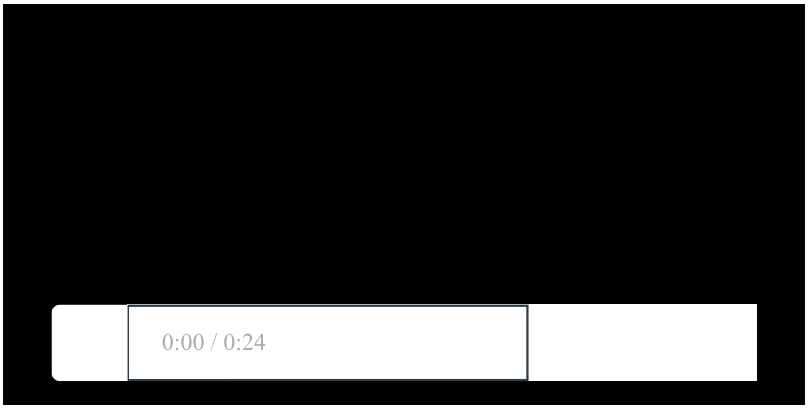
Air vibration, Phoneme, Sound of a word, Sound of a full sentence

Etc.

When I segregated these concepts above in a bucket like "Sound of words" or "Textures and Patterns" or "Sentences", I was creating something that we know as a "Block" of layers. Every layer would have 100s of thousands of channels, where each channel is responsible only for one thing. For instance:

At some **higher-level blocks**, a few of the channels could be vocals, guitar, bass , synth, and drums.

When we think in terms of channels, we can "split" the song being played above into its individual components, like the sound made by piano, guitar, bass, drums, etc. When everything comes together we get magic!



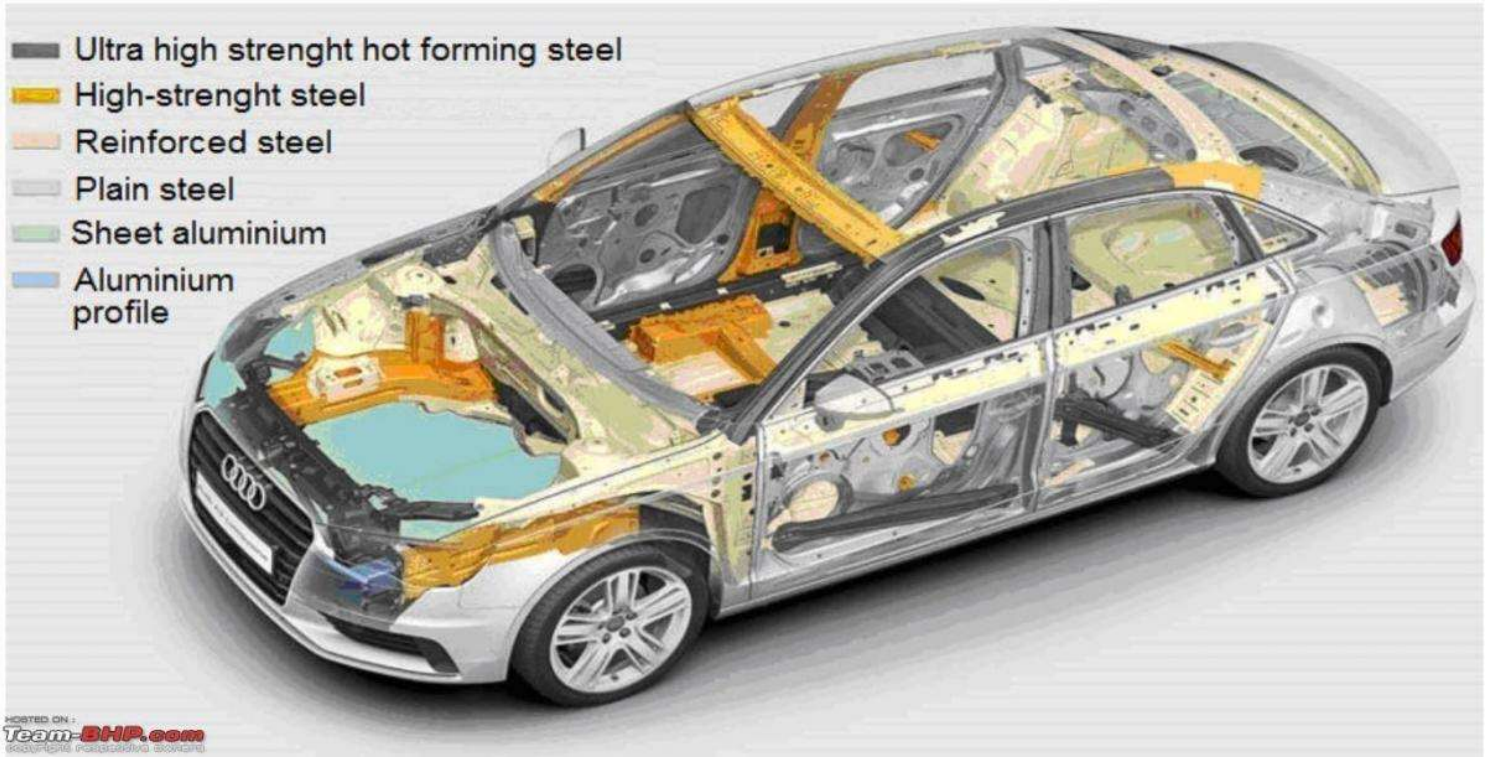
In the image below, one of the "lowest-level blocks" could be the alphabet. What is the maximum number of channels possible in this block (assuming the size of the characters does not matter)?

Typography Art

In the image below, a somewhat mid-level block could be represented by the ingredients:

Ingredients for Rice Pulao

Based on the concept, these channels (individual concepts/features) would be anything. Like material type in this image:



Car segmented based on its material.

and component level here:

All the components of the car

In both cases, we're using different kinds of channels, but I want you to be convinced that in the former image, channels were more "basic" than in the latter image. The later image had more "complex" channels.

What did we learn till now?

Anything complex can be made from simpler stuff!

Well, that is what AI is all about. How do we break down the problem into the simplest possible stuff, that can then be combined in trillions of ways to represent anything that we may want?

Origin or The Story of Two Missing Cats

CAT 1

Image created on MidJourney.

CAT 2

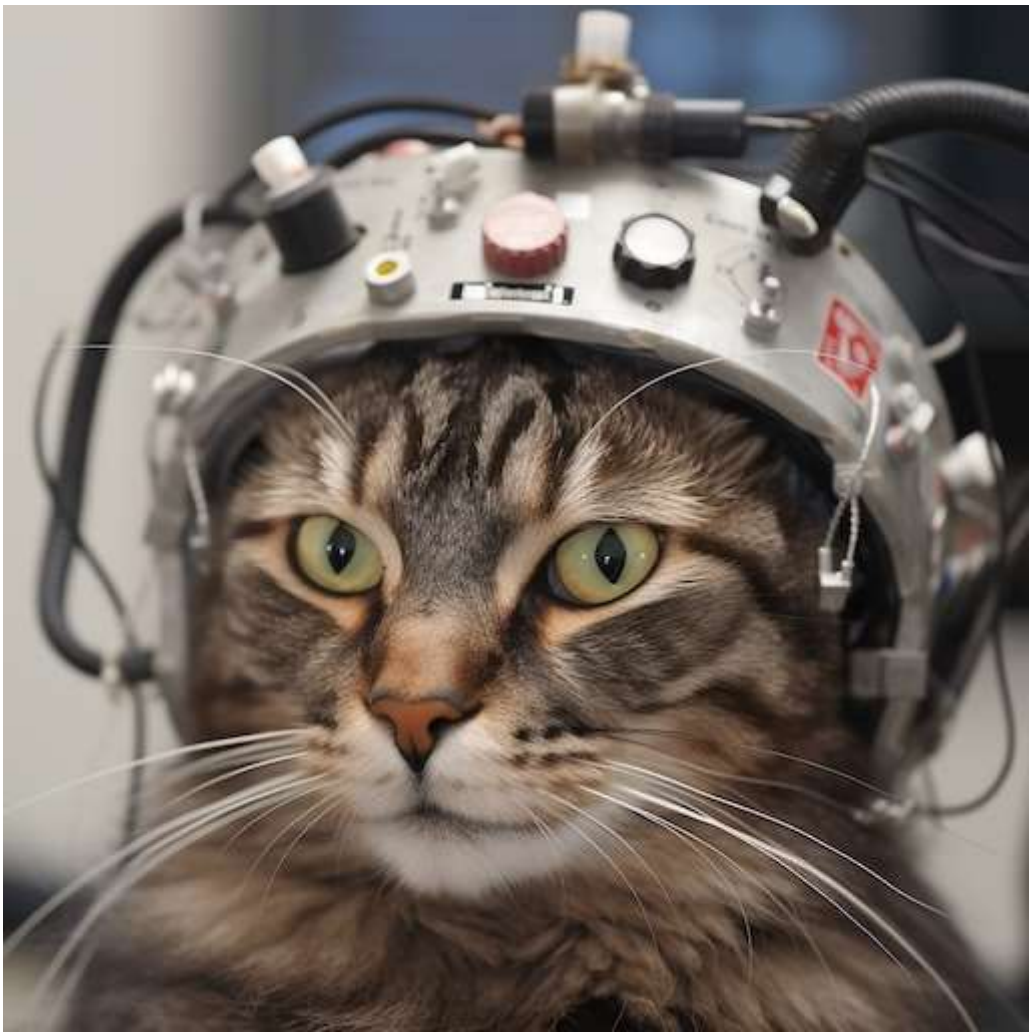
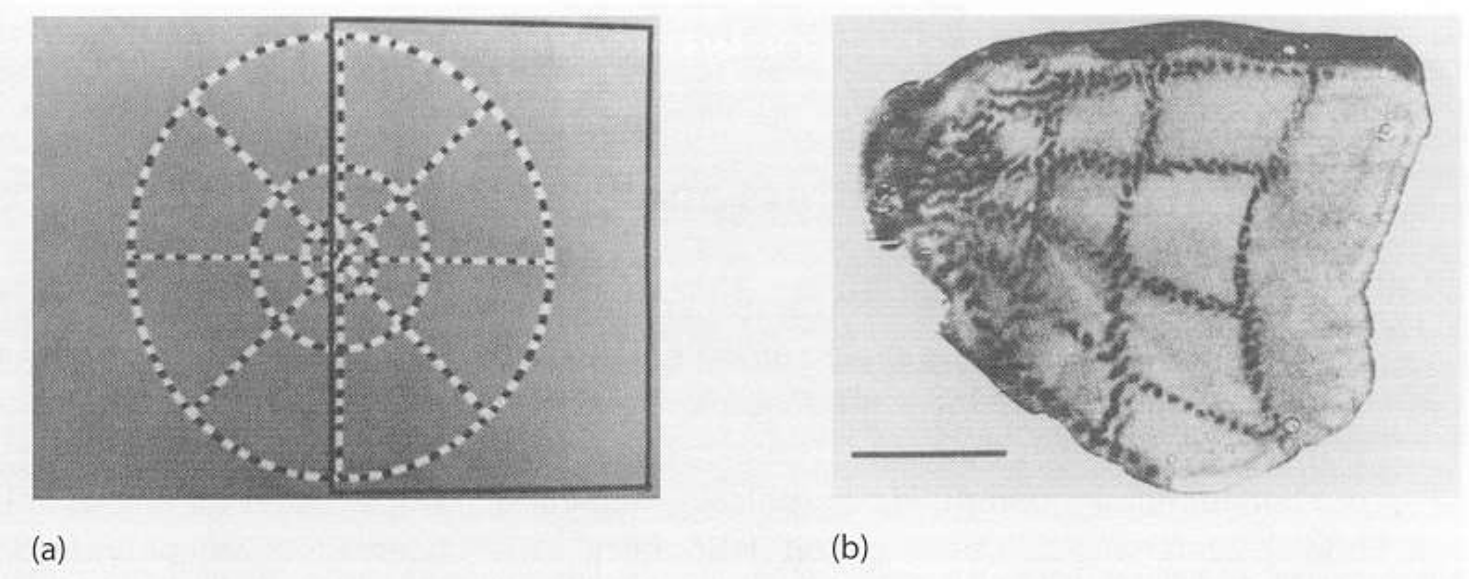


Image created on MidJourney.

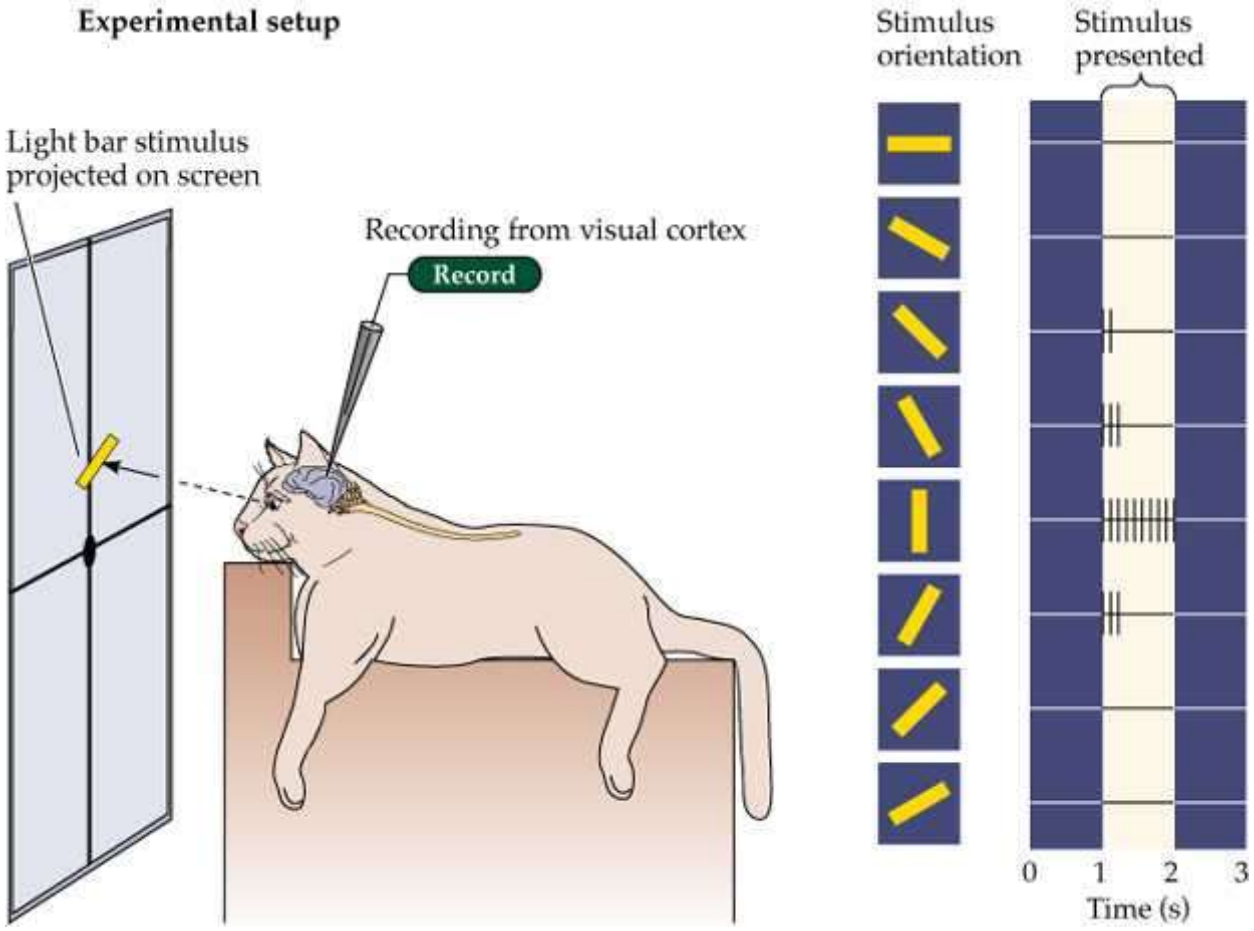
CAT 1: Experimental Results



If you are interested in learning more, read more about retinotopic maps and fMRI

When we see the image on the left, it gets "printed" on our brains, literally, as shown in the image on the right!

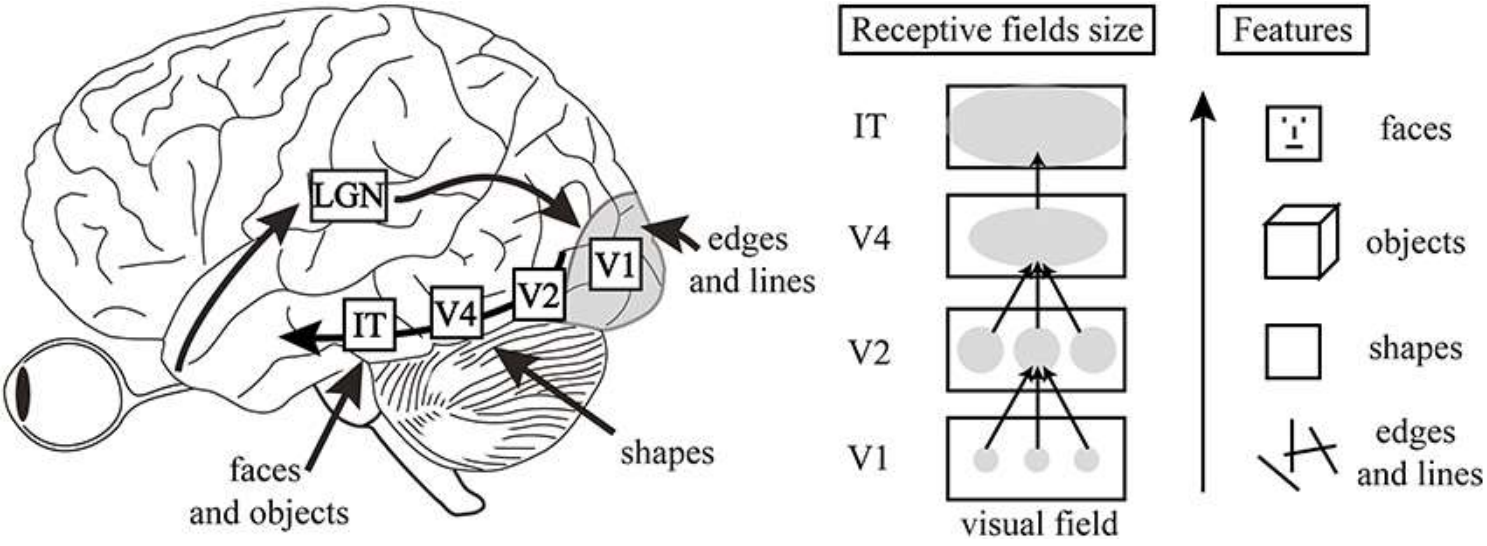
CAT 2: Experimental Results



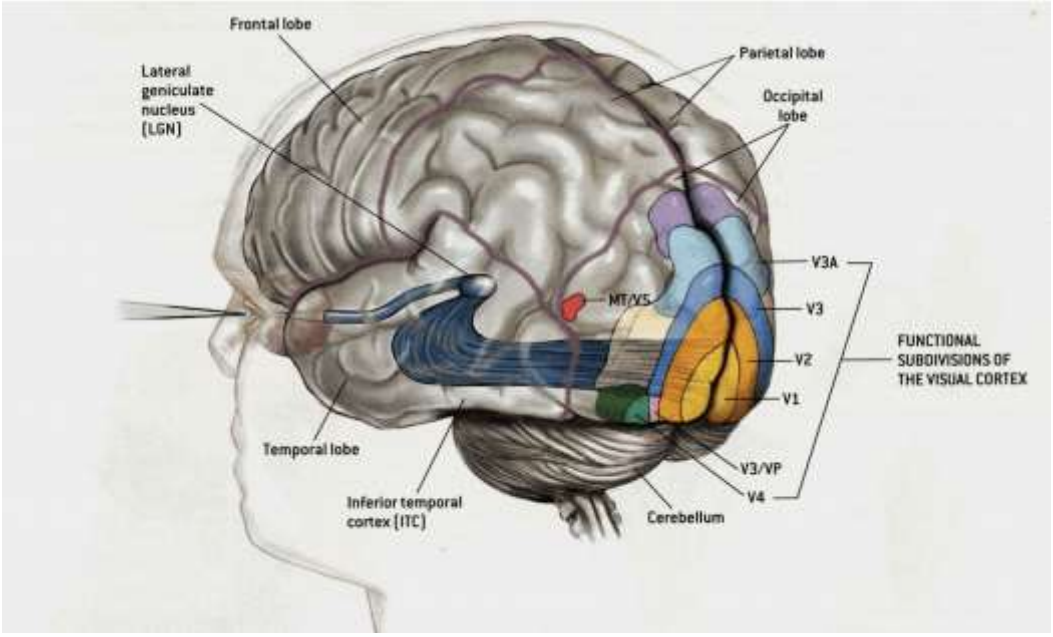
Would highly recommend you to look at this video and subscribe to its channel as well:

But what is a neural netwo...

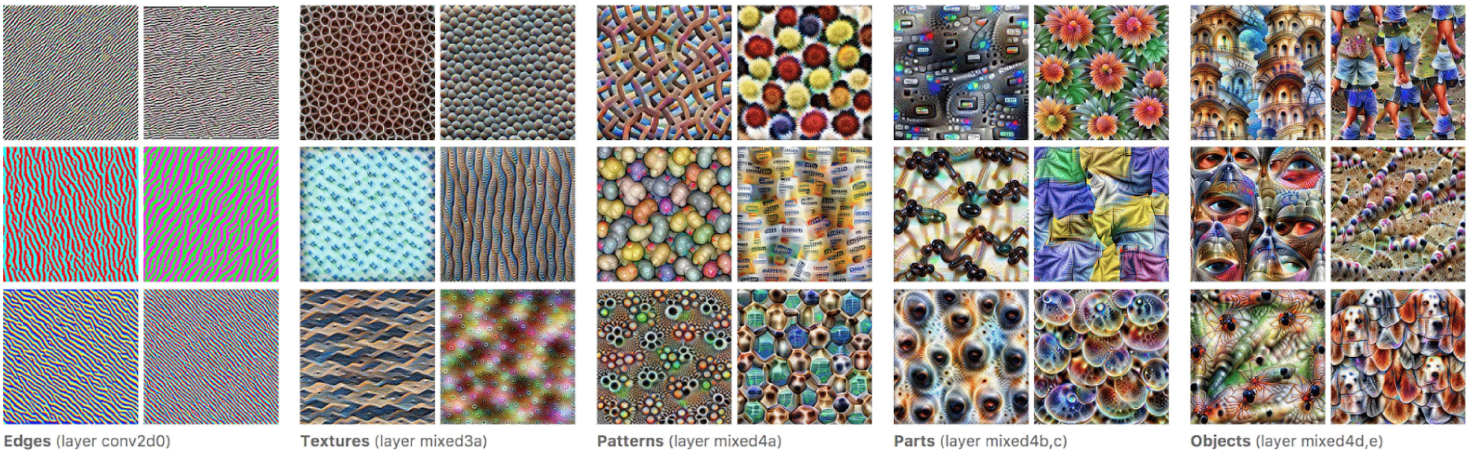
In our own visual cortex, we start with simple shapes (edges and gradients), combine them to make complex shapes (patterns/textures/part of objects), and finally get to see (the objects or the scene)



If you are interested in the above topic, please read more on this ***link*** ***(https://www.researchgate.net/figure/Left-A-typical-hierarchical-feedforward-model-where-information-processing-starts-at_fig1_267872860)*** .

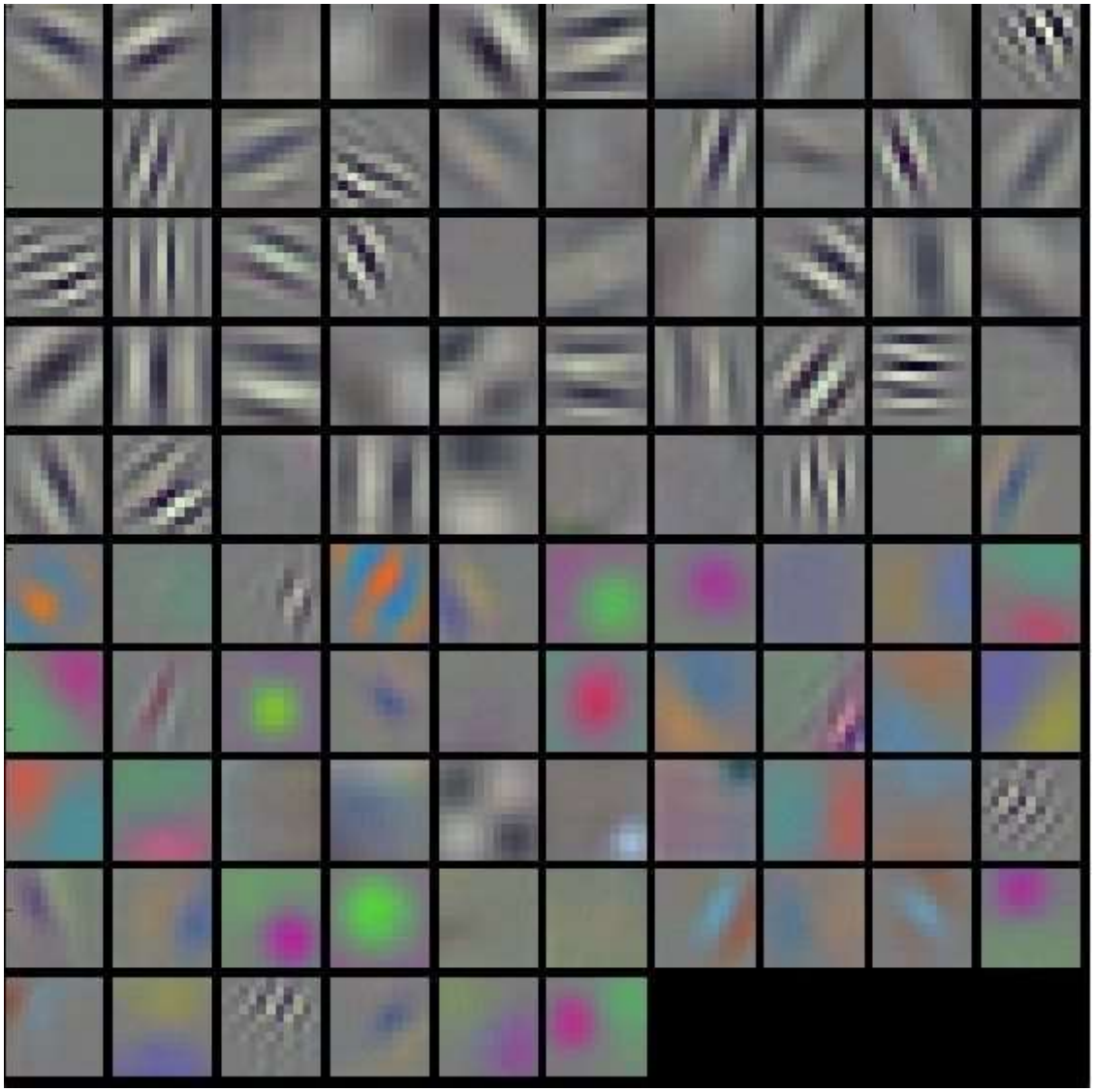


Something dramatically similar happens in the deep neural networks! And being artificial, we can actually get them to show us what happens inside them!



A Kernel is a feature extractor, and these feature extractors are what is learned in a DNN. Above you see an "extrapolation" of the feature a kernel extracts as if the whole image was just that feature.

We can even query the very first "block" of the DNN and see individual features of each "kernel" extract:



Above what you see is what a kernel or feature extractor or learned matrix exacts. Since this kernel was 11x11 in size it can extract only 11x11 features. If we extrapolate this 11x11 onto a big image, we'll see something similar to the image above.

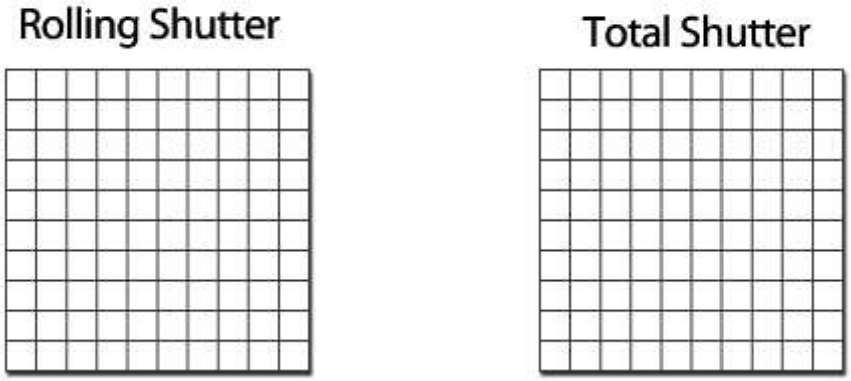
Let's look at this time-lapse to appreciate how simple strokes can make something really beautiful. Here we spend a few minutes, while our brain-eye system will spend less than 100ms

Again, the concept we are stying to learn here is that *complex structures or things can be built from simple strokes,* similar to the fact that we are communicating right now using just 26 alphabets!

Core Concepts

We need to understand a few other concepts before we jump to Convolutions. One such concept can be understood from the difference between a Rolling Shutter Camera and a Global/Total Shutter Camera

The reason behind these effects is how the pixels are actually created.

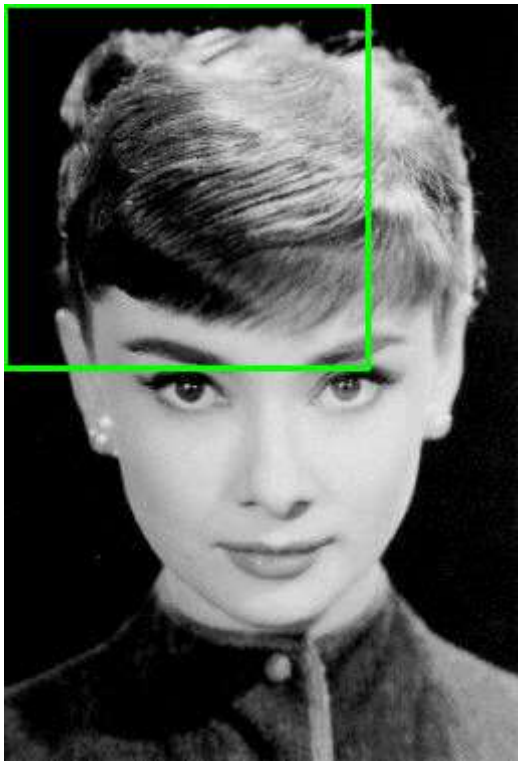


Rolling vs Global Shutter

Of course, Global Shutter cameras always make sense, but they are complex, expensive, and bulky to manufacture. All mobile phones use Rolling Shutter cameras, but now the speed of capture has increased a lot, hence we don't see these effects. Rolling Shutters are digital in nature, and Global Shutters are mechanical.

DNNs would use a rolling shutter kind of concept, but our brains use global shutters kind of capture.

A slightly extended concept was used during the Jurrasic era by computer vision engineers. Its called Sliding Windows



Sliding window

To actually implement a global shutter kind of processing in DNNs, we would need to change the hardware itself. Today hardware processing is based on "clock ticks" and we can process a small amount of data in a very very small tick

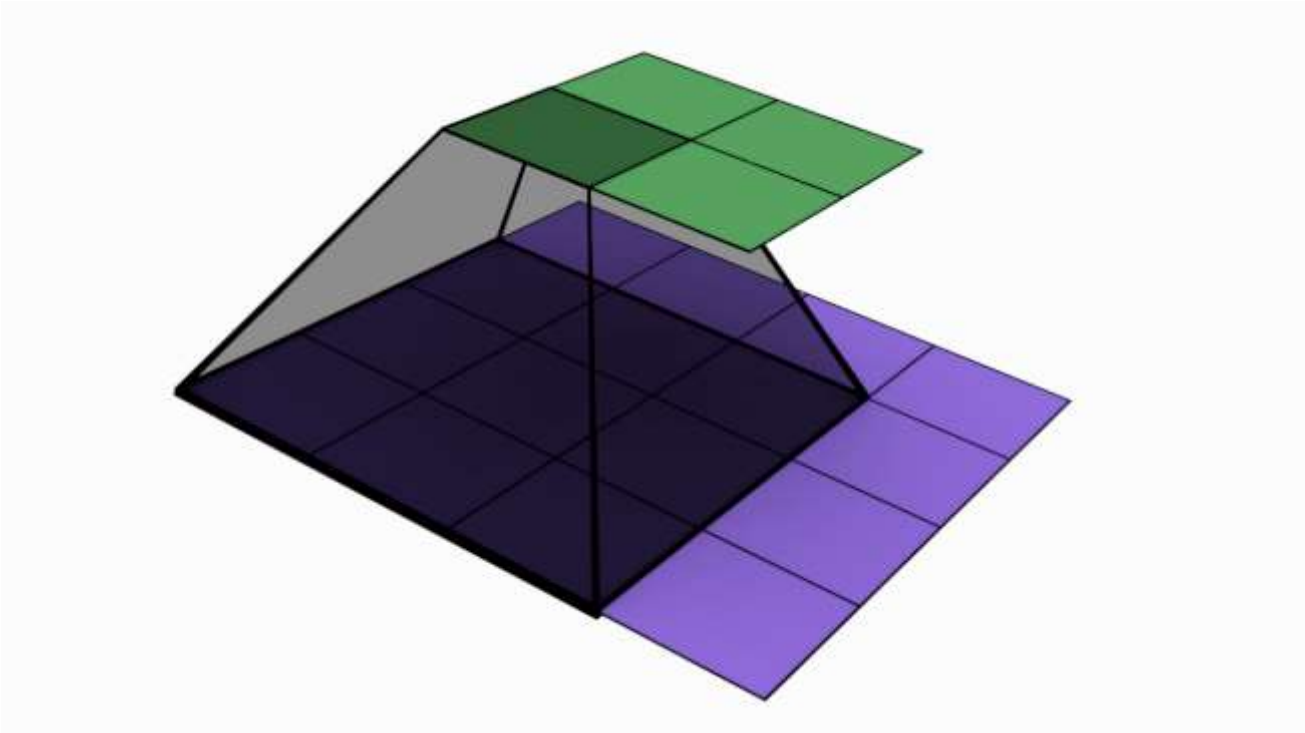
In both of these examples, let us focus on the "area in context". In the images above we either have a horizontal block or a square block of data that we are processing or "reading".

Now "reading" here basically means, that we are storing it for some kind of processing.

The pixels that are going to be stored will be worked upon by some algorithm, which ultimately would mean that some numbers would be multiplied/divided/added/etc to them.

These numbers are called kernels.

Convolutions

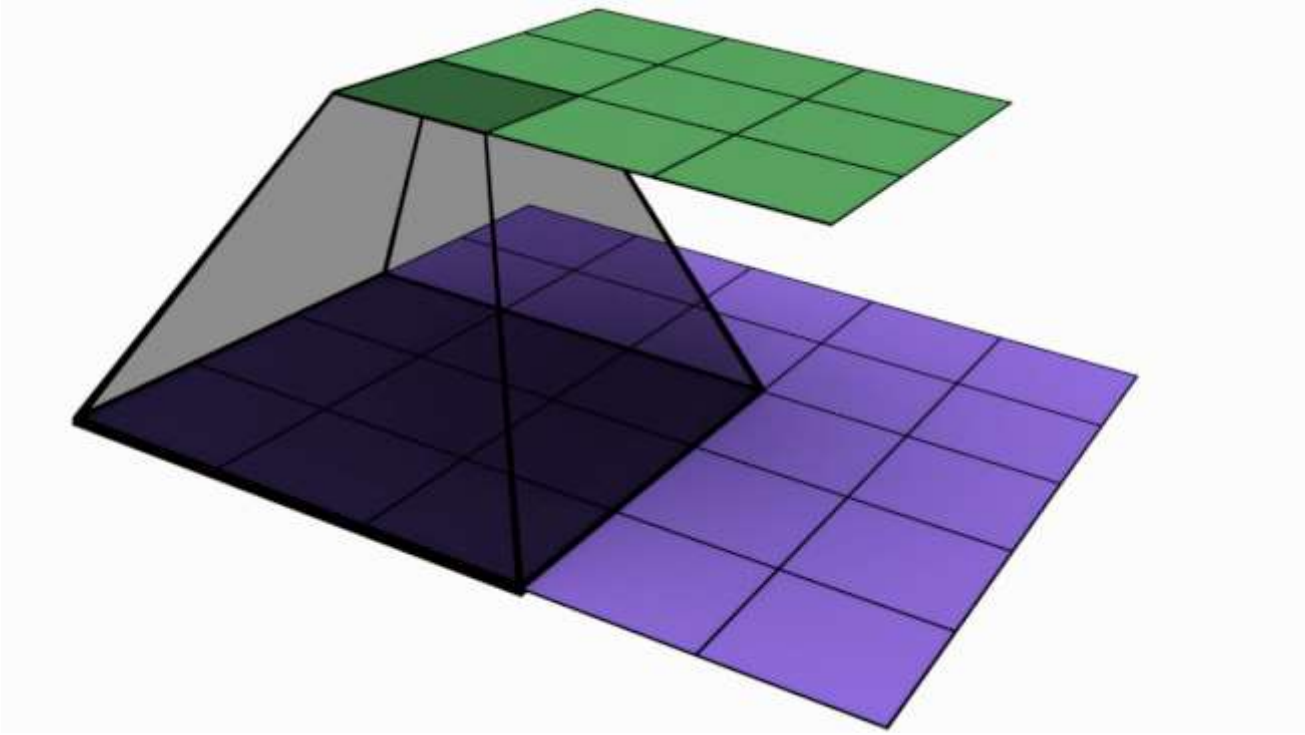


A 3x3 kernel on a 4x4 image.

Here we are "reading" 3x3 numbers on a 4x4 image. The moment we read these 3x3 pixels we multiply them by some 3x3 other numbers (identified by a DNN). These "other 3x3 numbers" are called Kernels.

Let's check this quickly (https://codepen.io/wallat/embed/yLymMey?)

So, there exist simple 3x3 matrix numbers that can easily identify basic lines/edges.



A 3x3 kernel on a 5x5 image.

So if we convolve a 3x3 kernel on a 5x5 image, the output we would create will have a resolution of 3x3. This is true only when:

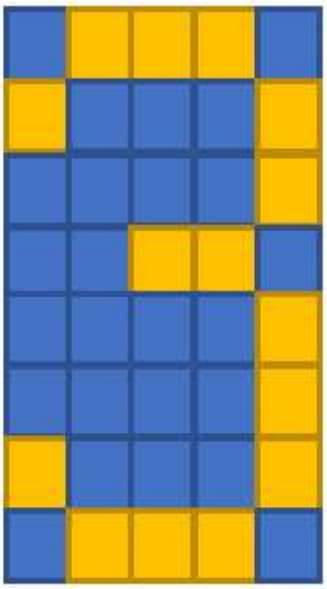
- we are not going outside of the image (by adding imaginary numbers). This by the way is called padding. We can add imaginary black/white (0/1/255) numbers and then allow our kernel to slide out of the image (why would we want to do this>>Session 3)
- we are not using a stride of more than 1, i.e. we will cover each 3x3 section immediately after 1 pixel

In the images above, our kernel skips/jumps only 1 pixel. If we were to jump/skip 2 pixels, then our kernel has a stride of 2 (pixels).

Fully Connected Layers

Convolutions are fairly new compared to Fully Connected Layers, and quickly took over, the moment we had enough computing because FC layers suffered from a serious issue (which transformers solved!)

Let's look at this image:

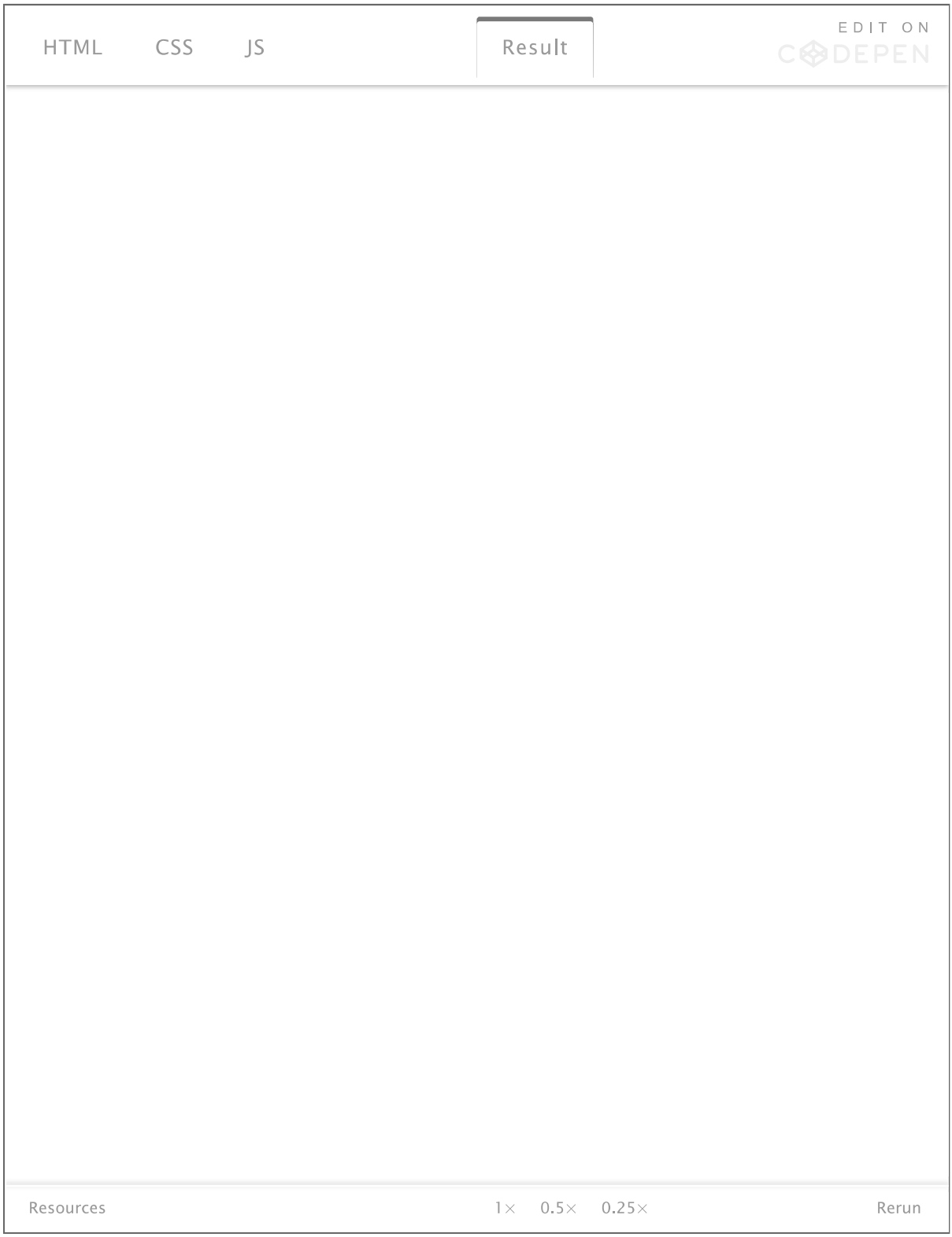


Digit 3: represented by 5x8 pixels

We can unroll this image and make something like this:

Same Digit 3: represented by 40x1 pixels.

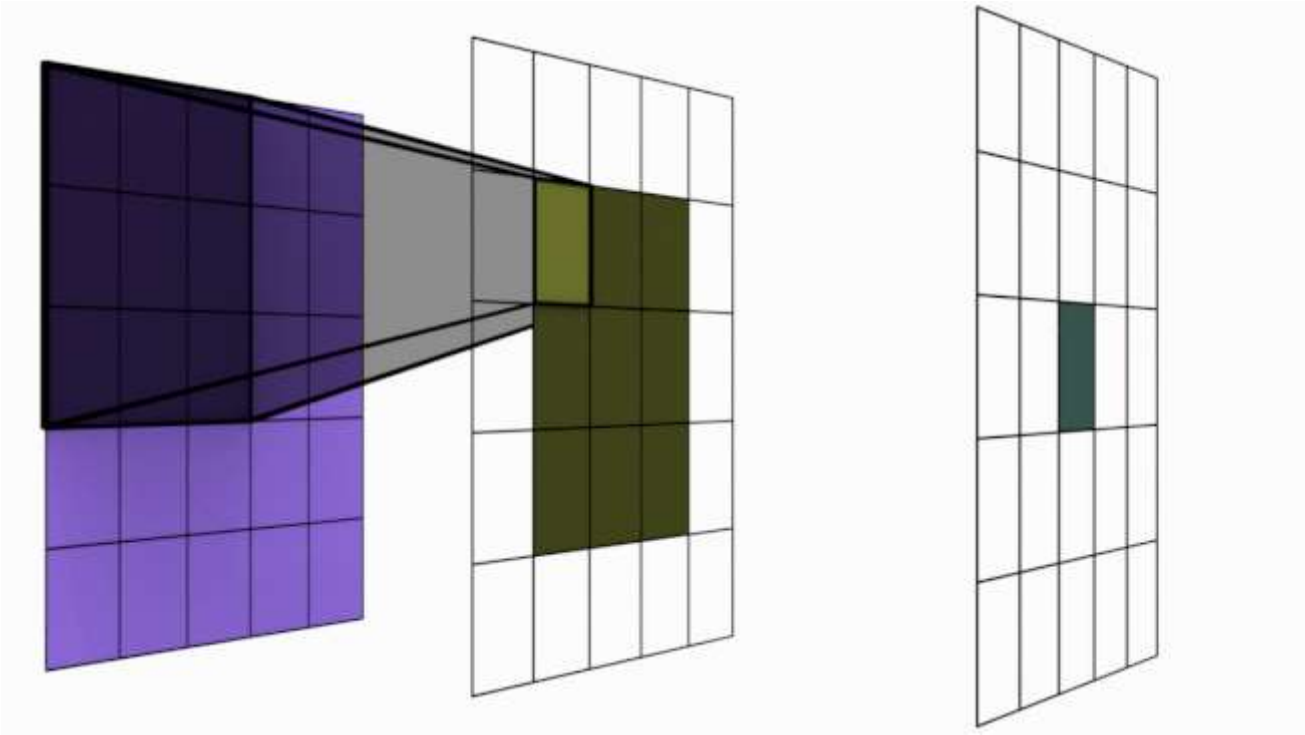
And now instead of reading 3x3 2D data, we can read this 1D data, where every number is multiplied by other numbers as shown in this example:



More about these in the next session.

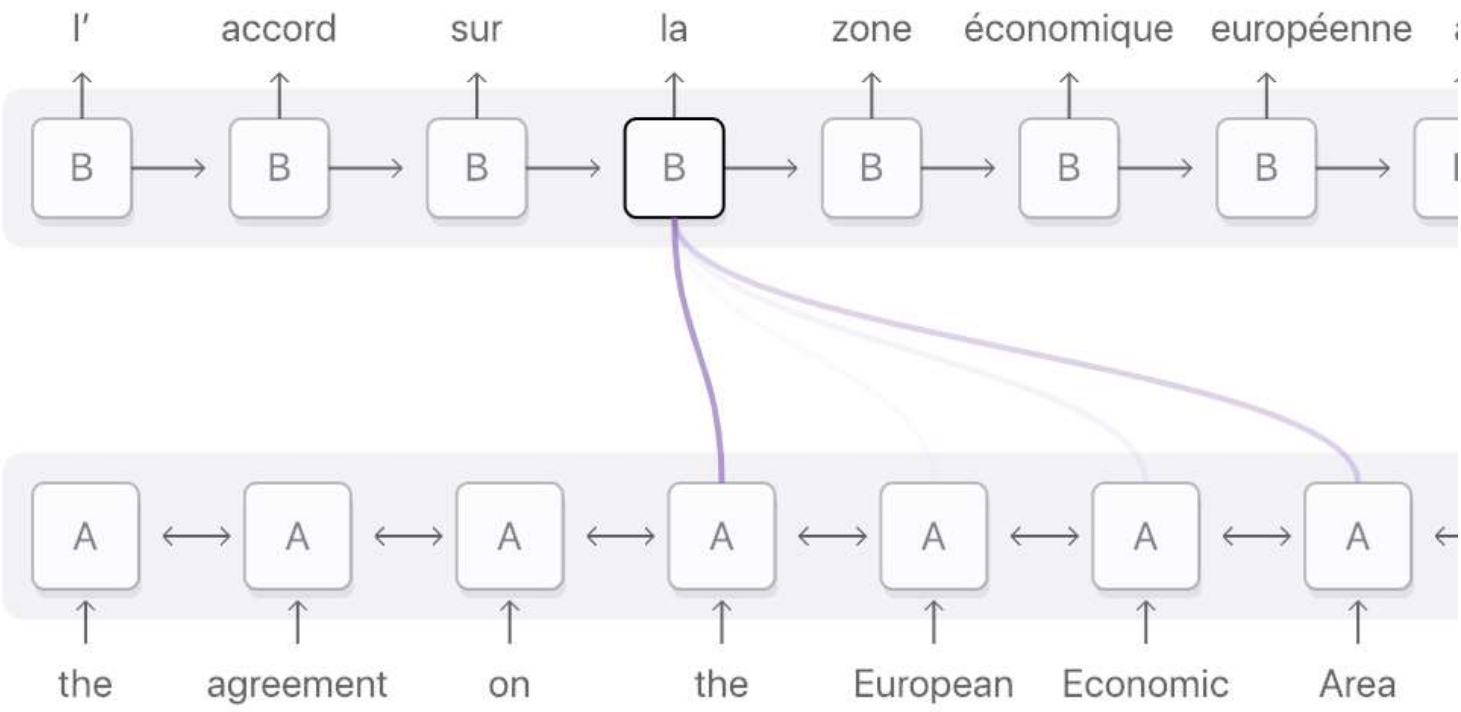
Receptive Fields and Attention

The most important image in ERA for Vision:



Receptive Fields

The most important image in ERA for NLP/Audio:



Let's check the ***source*** ***(https://distill.pub/2016/augmented-rnns/#attentional-interfaces)*** *for animation.*

Important Note on Vision vs NLP.

In the case of the language, we have broken down concepts into words. Each word means something, and based on the context (bank example) we know what it means. In fact, we (sort of) have a fixed exhaustive dictionary of words that we can download and make GPT4 out of it!

In the world of 3D graphics, we have done the same thing. We have these edges, that make polygons. Using these polygons we make primitives like spheres, cubes, etc., and complex shapes.

But we are (not all) graphics designers and we don't have the intuitions to break down the big picture into smaller pieces.

But for a moment let's imagine that we indeed had characters and words like concepts in vision. The next step would have been to directly use them to define the image/world (just like in NLP). And now you immediately start seeing where exactly the architectures for Vision and NLP would meet!

The convolutions we saw above would help us create these vision words (edges, gradients, patterns), and then Fully Connected Layers (Transformers) would help us connect them with language in a single model!

The Assignment

1. Your quiz questions are on a separate quiz listed as "S1 Quiz". We do not have any assignments for Session 1
2. 100 Pts
3. 0.99 Weeks

Videos

Studio

Google Meet