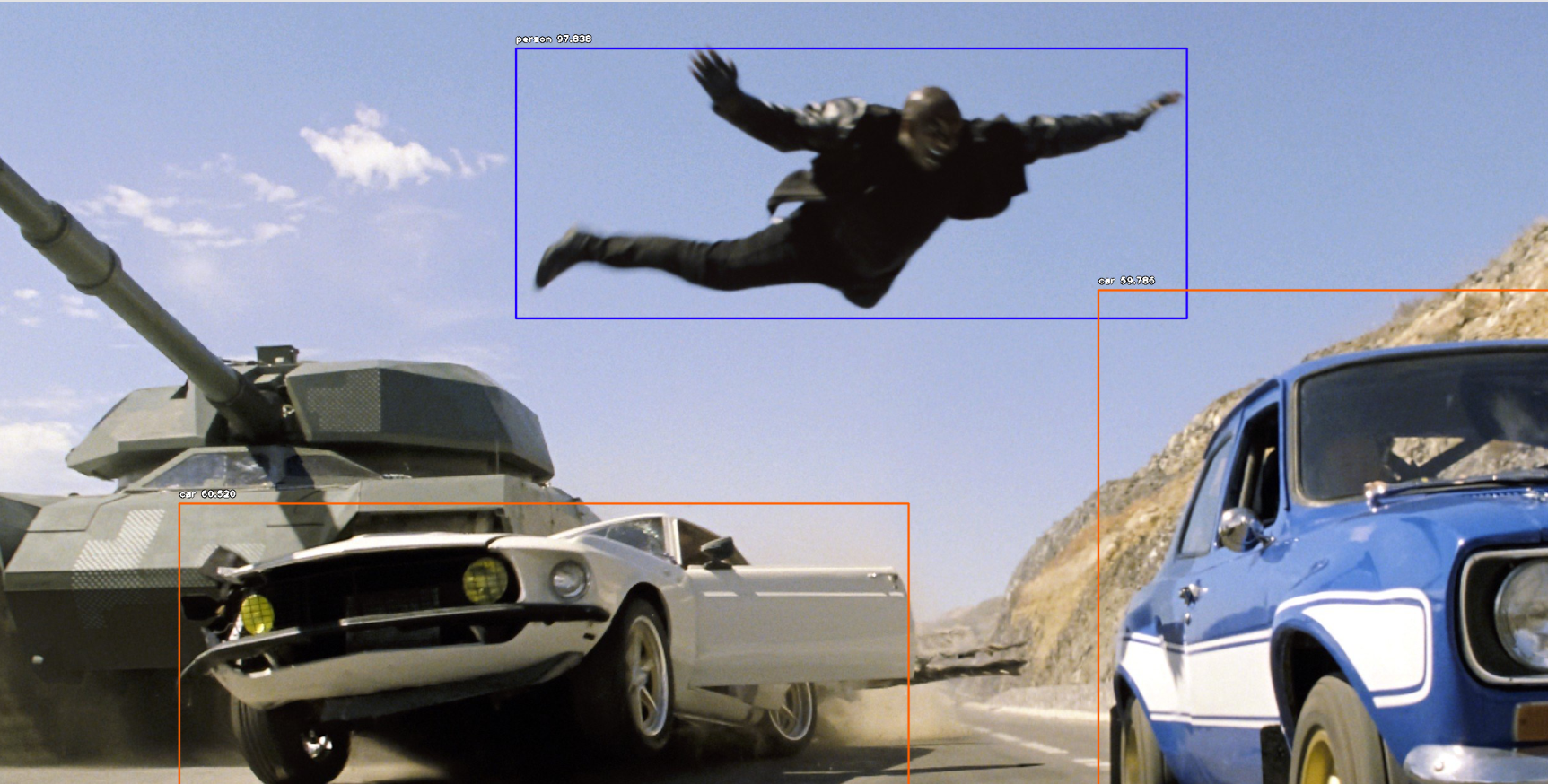
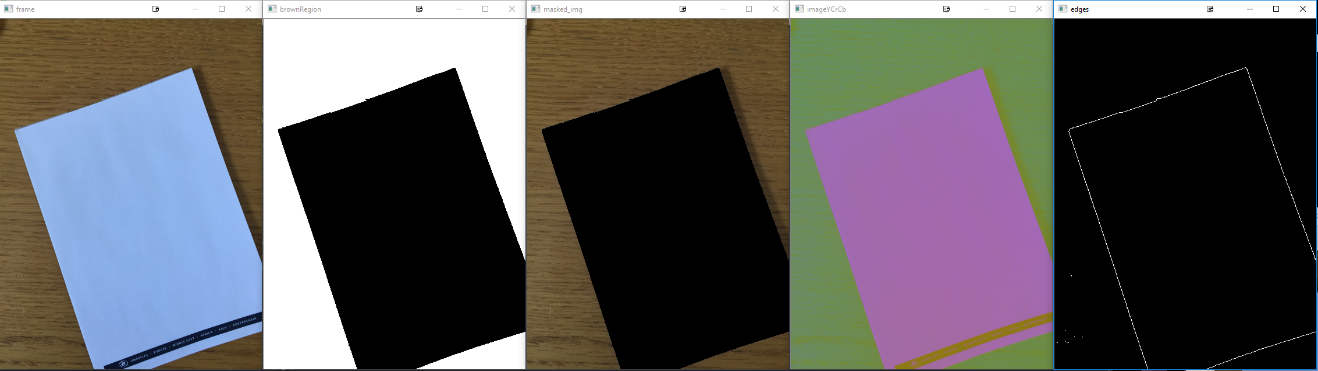
**An overview of Computer Vision**

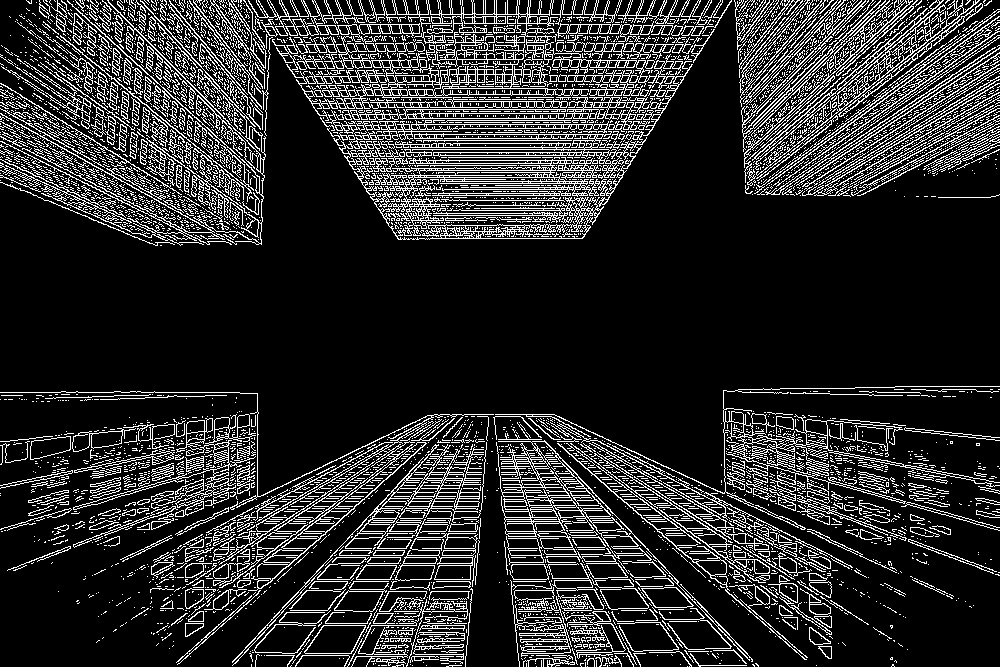
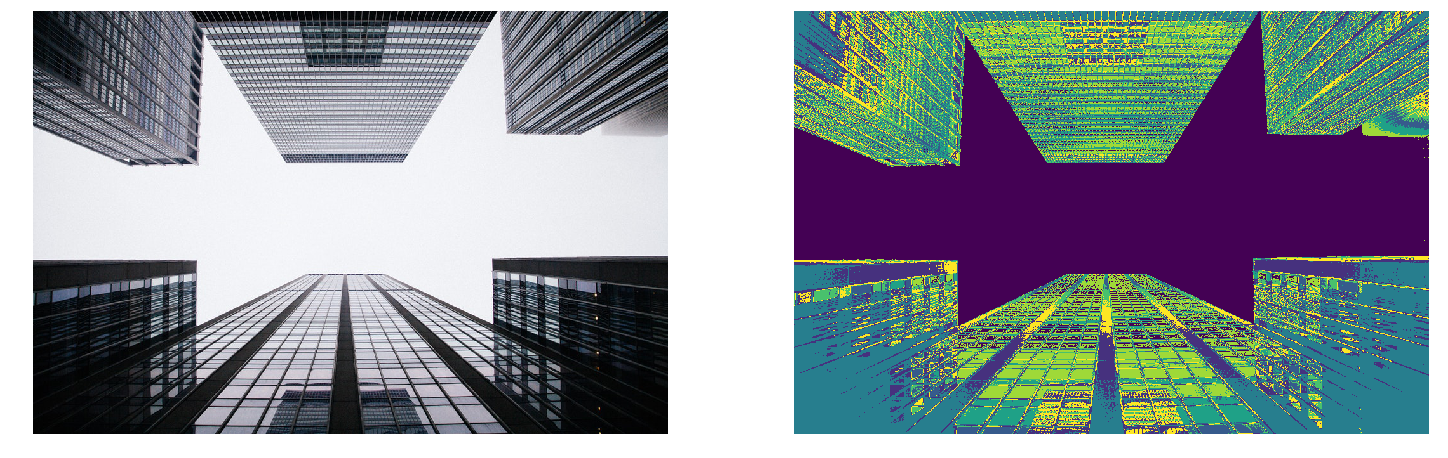


Computer Vision is a subfield of Machine Learning that aims, as the name suggests, to help computers “see”. If that came as a surprise, then consider that computers don’t even “see” as we do. That *is* our end goal, but till then we just rely on techniques that help other algorithms detect patterns.



Let’s look at these buildings; a pretty interesting view. An artist might notice the perspective, a civilian engineer might notice distances between buildings, someone else might notice the reflection, and so on. All a computer sees is a bunch of zeros and ones (and often some numbers in-between). By tricking the computer a bit, instead of 0 and 1 we get numbers between 0 and 255, and use their combinations to denote colors. We usually prefer the RGB format, which stands for Red, Green, Blue. An RGB value of (255,0,0) is pure red, while (0,255,0) is pure green. The value (0,0,0) denotes black, and (255,255,255) denotes white (remembering our art classes, aren’t we?). Each pixel in the image is just a very small box that contains one such triplet. This triplet contains what we call 3 color channels (Red channel, Green channel, Blue channel). Have you ever seen a white-and-black image? These are just images with only one channel, and we call them “grayscale” because they can be thought of as shades of gray (too dark and you have black, too light and you have white). So far so good.

But just because computers view a bunch of numbers that does not help in answering a question like how many different objects there are in a picture, or what is the shape or size of that object. The algorithms that experts created over the past decades were aiming that just that, but they couldn’t work with big, colored images due to computational constraints. Also, since computers are not really good at thinking, we need to help them. That’s where feature selection/engineering comes in. Instead of using an image, we first create some transformation of the image so that we can both reduce the computational needs and help computers view only what is important. One such feature engineering method is Edge Detection (which detects points where there is a lighting difference in the left-right or up-down axis) which you can see to the above. Roughly explained, it works by computing gradients (yes, almost the same stuff you used in high-school). There are many possibly ways to extract useful features (find corners, outlines, shapes), but we’re not going through them here. If you are interested, you can look up Morphology.



But there are still more cool stuff we can do! Image Segmentation is the process of dividing an image into parts. Above you can see how a very simple algorithm, KMeans Clustering, decided to divide the image above. You can see that it correctly identified the sky as one part of the image, and then divided the buildings into multiple more parts, all of which are very similar to each other (more similarly colored means closer resemblance).

Then, there are much cooler use cases. Using Multiple-View Geometry, and some pretty having math, we can use two cameras to create a 3D reconstruction (i.e. estimate depth) of a view, with ~$15. Take that, Kinnect! (Well, not really since accuracy will be a bit lower, and I’m not much of a math person)

We still have Virtual and Augmented Reality, Body Pose Estimation, Object Classification Detection, Face Recognition. All of them have different purposes but are under Computer Vision. It is a truly vast and interesting field. Before closing, we should mention Deep Learning which has invaded the space of Computer Vision, probably much more effective than any other field, and it has taken the lead in many different tasks, but the most revolutionary may be object detection:

