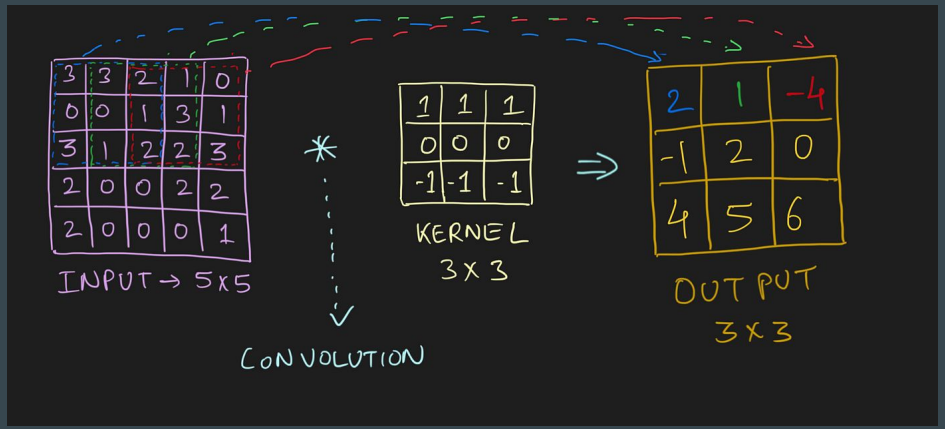
Convolutional Neural Networks

...

(and some basic computer vision stuff)

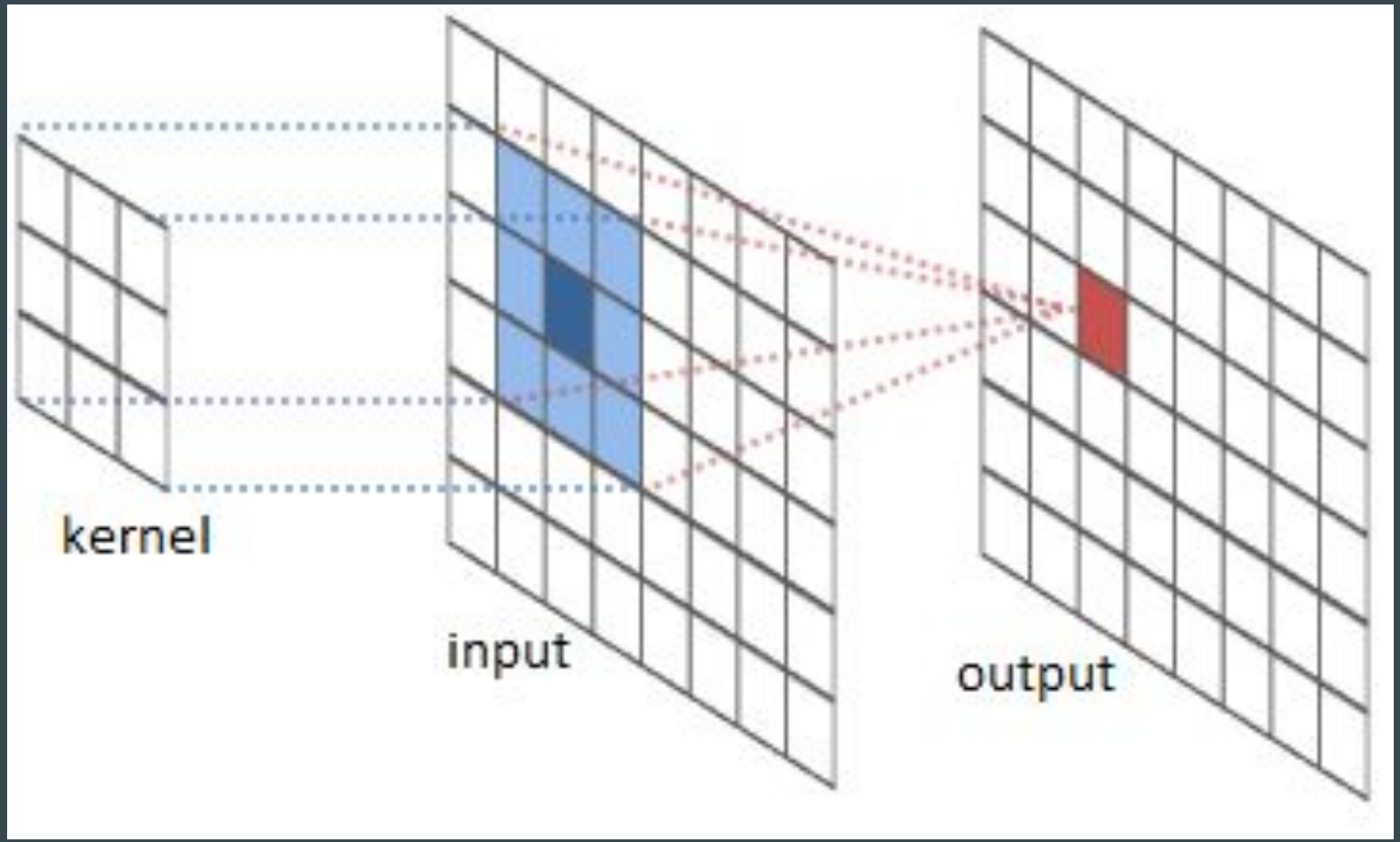
What Is A Convolution?

- From a computer's perspective, all images can be read as a 3D array (height, width, RGB values)
- Convolutions (from a computer vision application) helps us simplify this 3D array



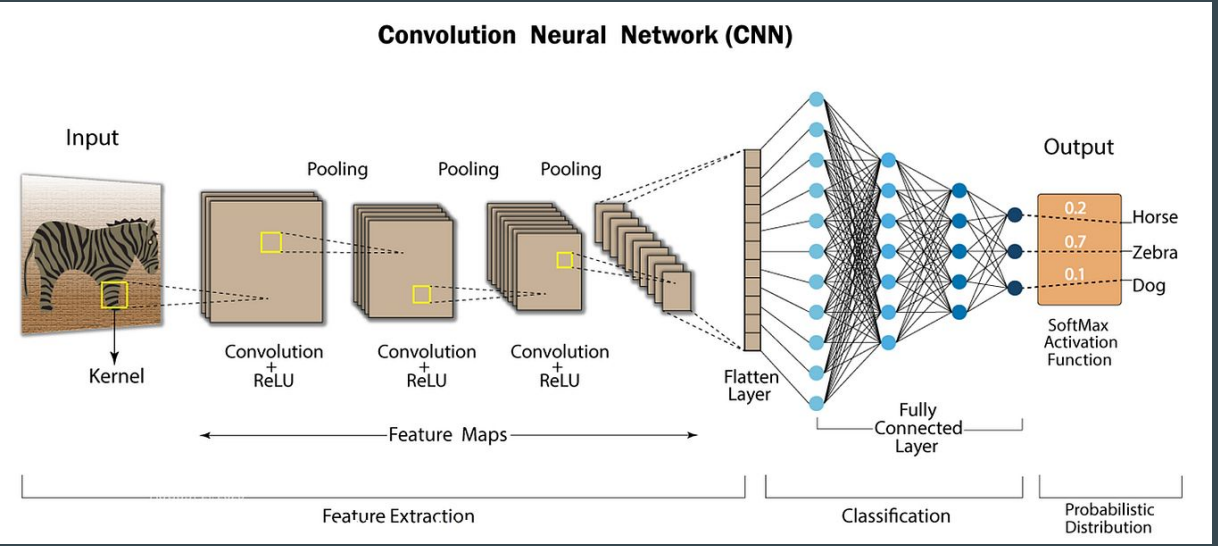
$$(f * g)(t) \overset{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau) g(t - \tau)\, d\tau$$

To convolve a kernel with an input signal:
flip the signal, move to the desired time,
and accumulate every interaction with the kernel


kernel
input
output

Convolutional Neural Networks

- Utilize a convolution to capture patterns
  - The output of a convolution allows the computer to look at small parts at a time, allowing the computer to find local features
- “Layering” these convolutions allows the computer to find more complex features



Pre-Processing

How do we turn an image into an array that can even be fed into a CNN?

- Our Objective: Make the computer's job as easy as possible
- Find ways to simplify the image for the computer
  - Data Transformation
  - Data Augmentation

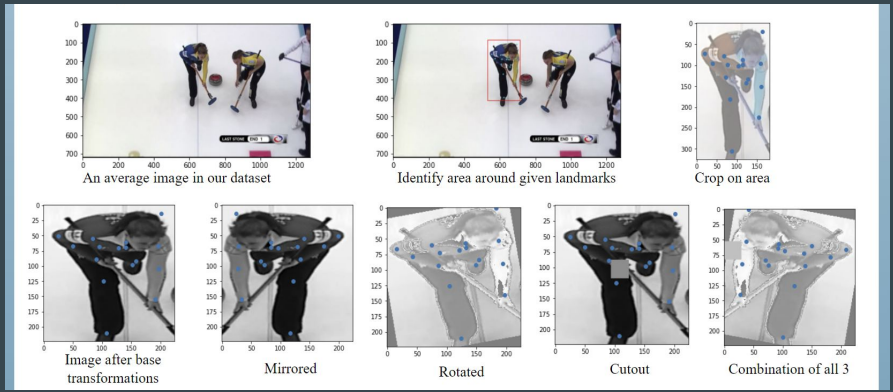
Data Transformation

- Crop onto your targets
  - Reduce the overall image by cropping onto what you want to work
- Get rid of color whenever possible
  - Reduces complexity of the images drastically from a computer's perspective
  - (Height, Width, RGB Values)
- Make all the images the same size
  - Standard is to resize to a 128x128 image
  - Watchout for changes to landmarks and other classifiers that may be on the image

Data Augmentation

Tons of ways to augment images

- Mirroring, rotating, filtering, perspective changes, blurring, drop out, etc.
- Introduces randomness into the training data, making your model better at predicting irl, reducing overfitting
- Allows you to artificially increase your dataset



Different CNN Architectures

- ResNet
  - Uses residual connections (randomly skipping layers) which allows for training deeper models without suffering from vanishing gradient
- MobileNet
  - Designed for mobile devices by having reduced size while maintaining the majority of perfor
- U-Net
  - Designed for biomedical applications by combining low quality layers with high quality ones (think Xrays)

How To Get Started On Your Own

- Grab a dataset from Kaggle!
  - Dogs and cats
- GPU heavy work (and camera)
- If you lack the skill to create custom implementations, you can load them in
  - Pytorch has a ton of libraries!
- https://www.youtube.com/watch?v=Z_ikDlimN6A