Advanced Topics in Deep Learning

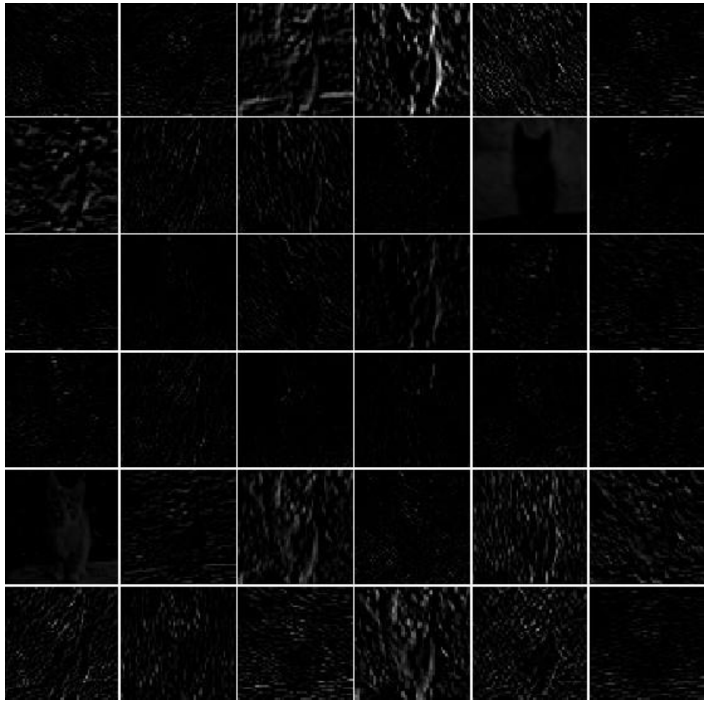
LEADINGINDIA.AI

NATIONWIDE AI SKILLING & RESEARCH INITIATIVE

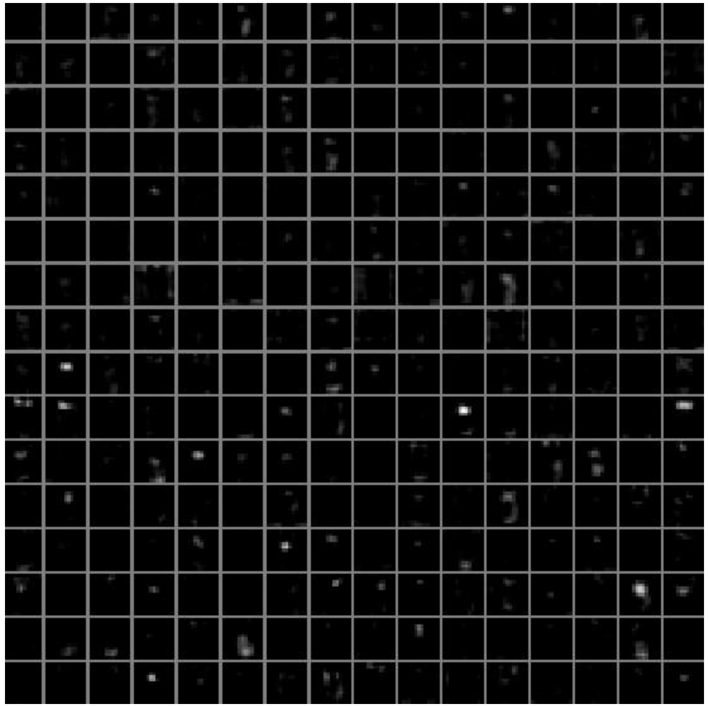
Layer Visualization using Activations

AlexNet Layer Activations

Input Image



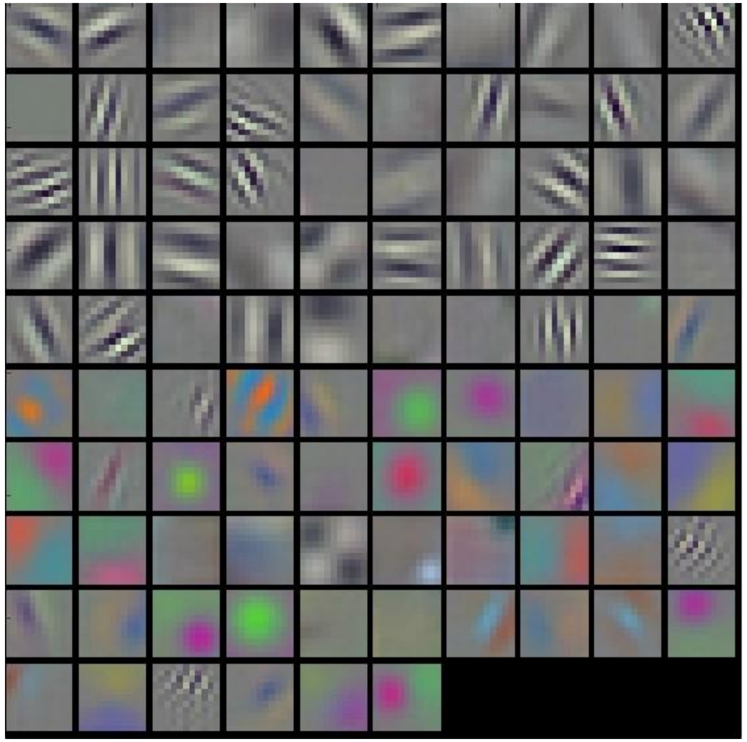
Activations on the 1st CONV layer



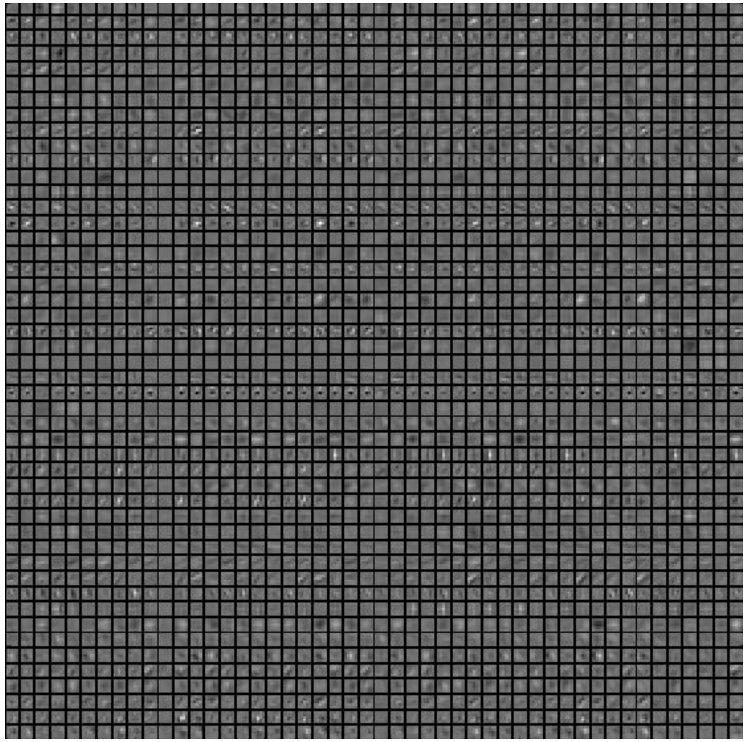
Activations on the 5th CONV layer

Visualizing the Filter Weights

Weights of 1st CONV layer

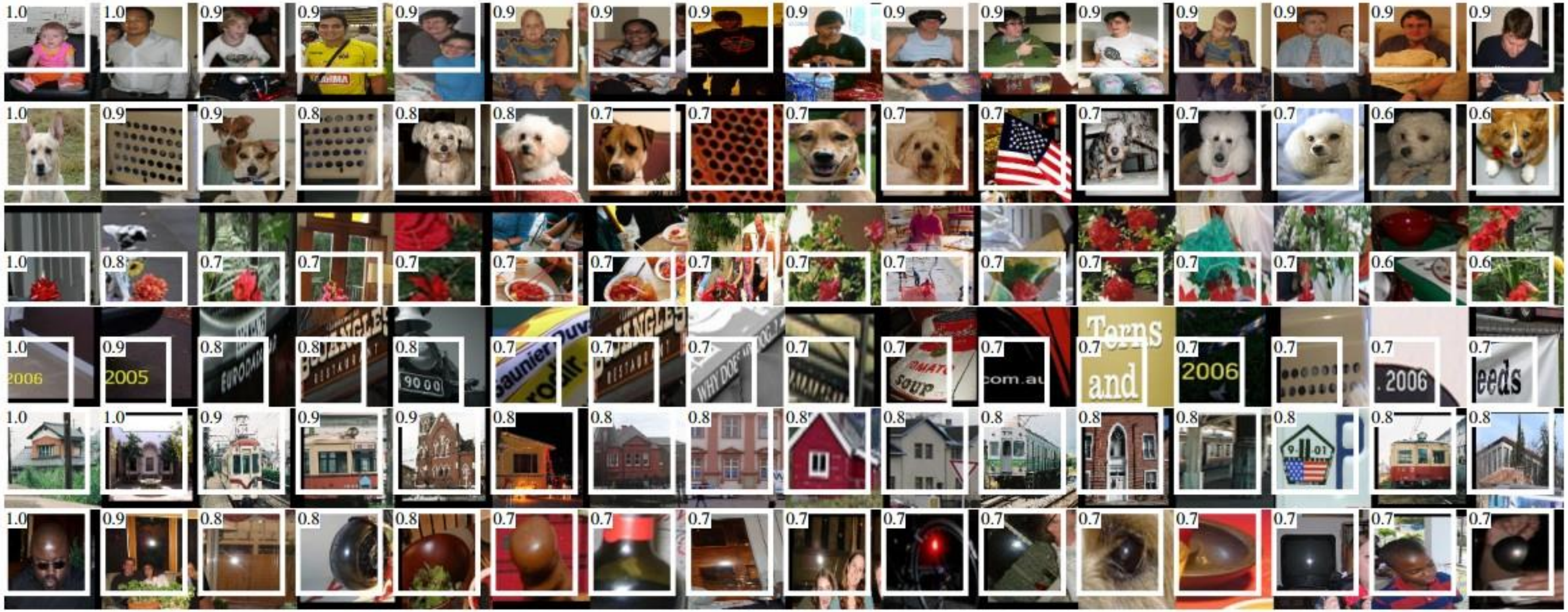


Weights of 2nd CONV layer



Notice that the first-layer weights are very nice and smooth, indicating nicely converged network. The 2nd CONV layer weights are not as interpretable, but it is apparent that they are still smooth, well-formed, and absent of noisy patterns.

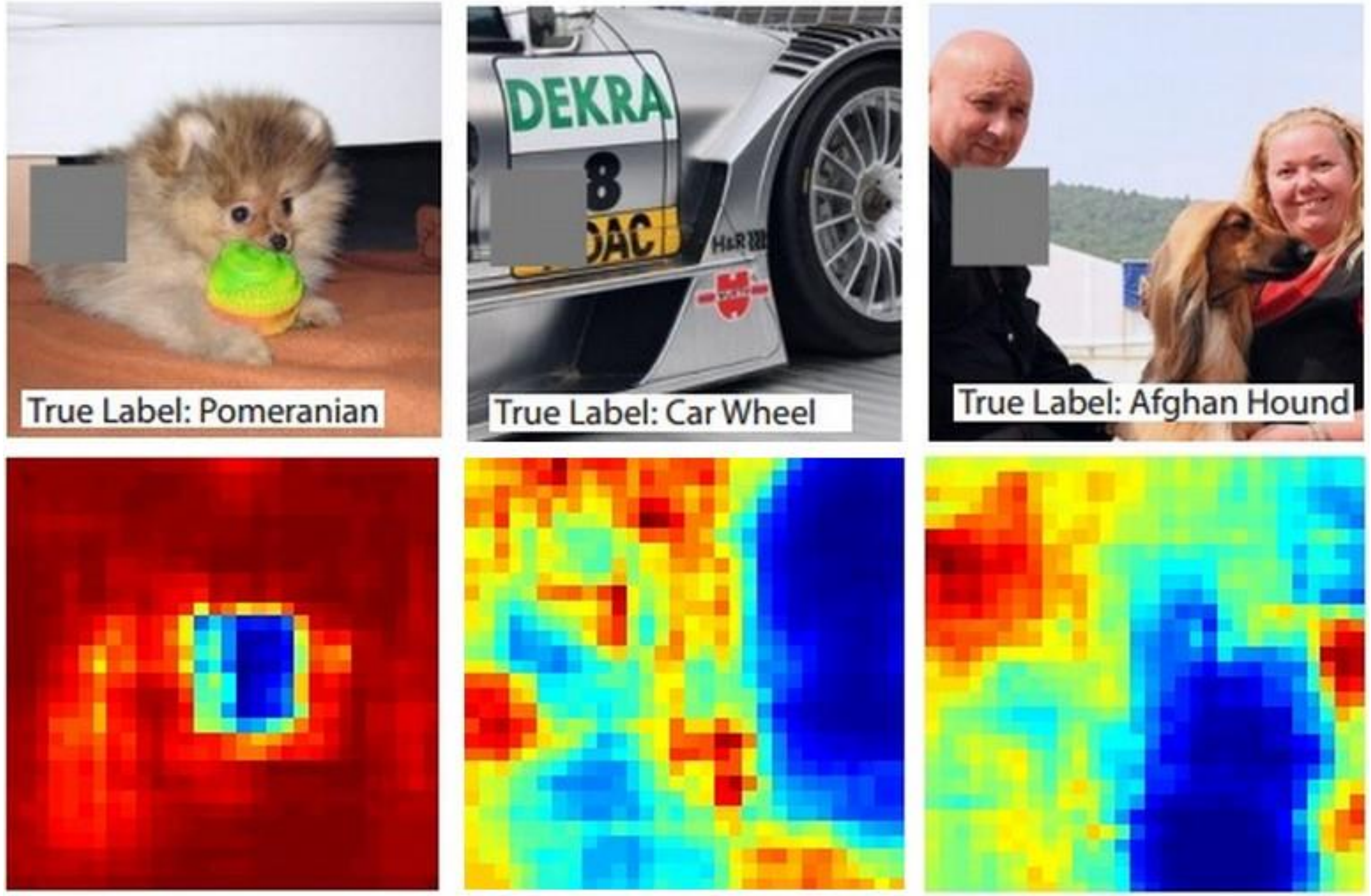
Retrieving images that maximally activate a neuron



Maximally activating images for some POOL5 (5th pool layer) neurons of an AlexNet

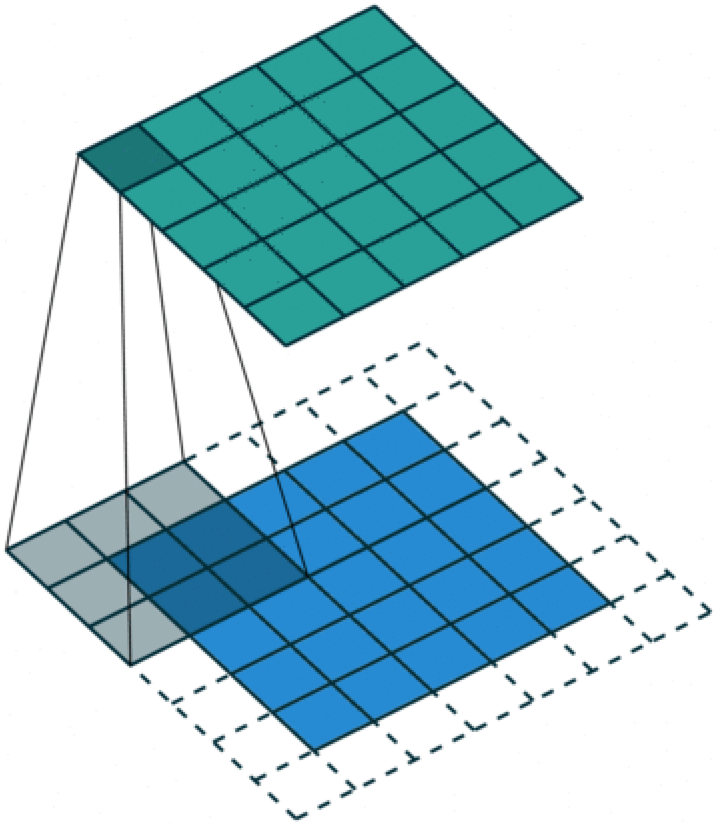
Occluding Parts of the Image

Three input images (top). Notice that the occluder region is shown in grey. As we slide the occluder over the image we record the probability of the correct class and then visualize it as a heatmap (shown below each image).

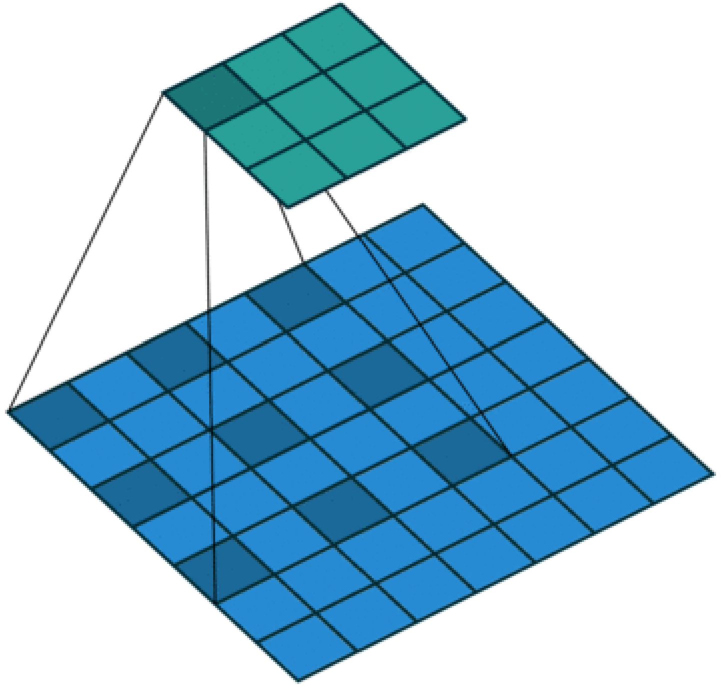


2D Convolution

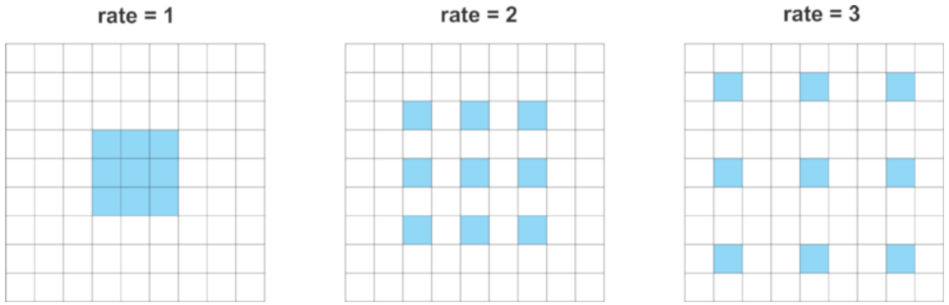
- **Kernel Size**: The kernel size defines the field of view of the convolution. A common choice for 2D is 3 i.e 3x3 pixels.
- **Stride**: The stride defines the step size of the kernel when traversing the image. While its default is usually 1, we can use a stride of 2 for downsampling an image similar to MaxPooling.
- **Padding**: The padding defines how the border of a sample is handled. A (half) padded convolution will keep the spatial output dimensions equal to the input, whereas unpadded convolutions will crop away some of the borders if the kernel is larger than 1.



Atrous/Dilated Convolution



- Dilated convolutions are particularly popular in the field of real-time segmentation.
- Dilated convolutions introduce another parameter to convolutional layers called the **dilation rate,** that defines a spacing between the values in a kernel.
- A 3x3 kernel with a dilation rate of 2 will have the same field of view as a 5x5 kernel, while only using 9 parameters. Imagine taking a 5x5 kernel and deleting every second column and row.
- Use them if you need a wide field of view and cannot afford multiple convolutions or larger kernels.



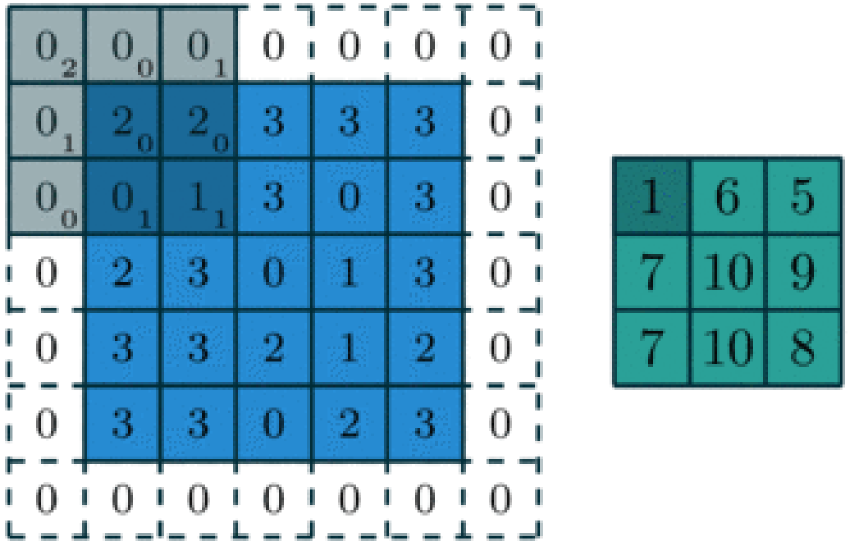
Dilated Convolution

- Dilated convolutions have generally improved performance in semantic segmentation results
- The architecture is based on the fact that dilated convolutions support exponential expansion of the receptive field without loss of resolution or coverage.
- Allows one to have larger receptive field with same computation and memory costs while also preserving resolution.
- Pooling and Strided Convolutions are similar concepts but both reduce the resolution.

3D CNN

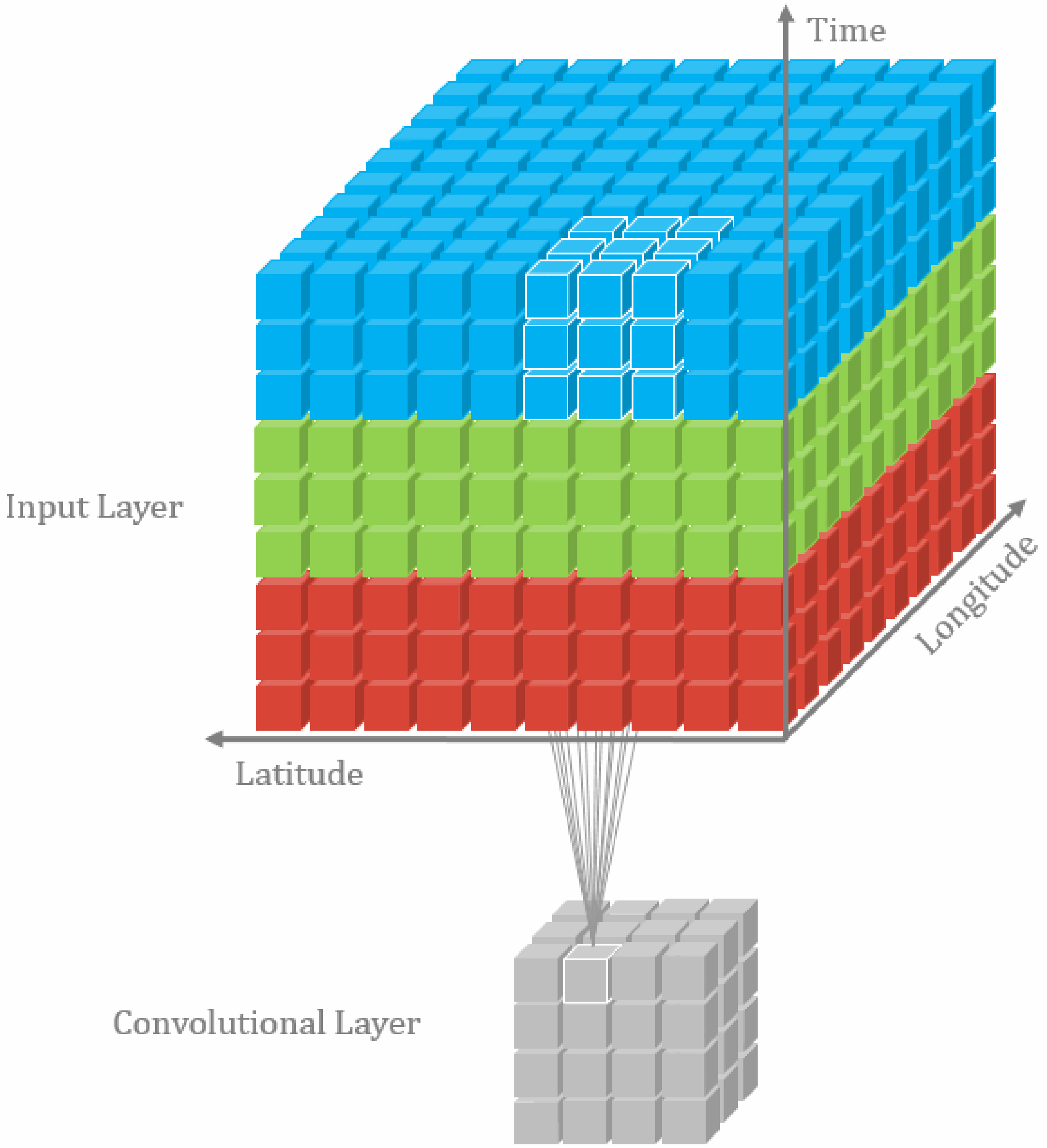
- 3D imageries such as MRI Scan, Human action in video sequence should be processed frame-by-frame
- Temporal property within the 3D imagery is important in extracting the features
- 2D convolution extracts features from only spatial dimensions
- 3D convolution operates on the input volume not only in the *X* and *Y* dimensions but also in *Z* dimension
- Building a 3D CNN requires, 3D Convolution as well as 3D pooling

2D vs 3D Convolution

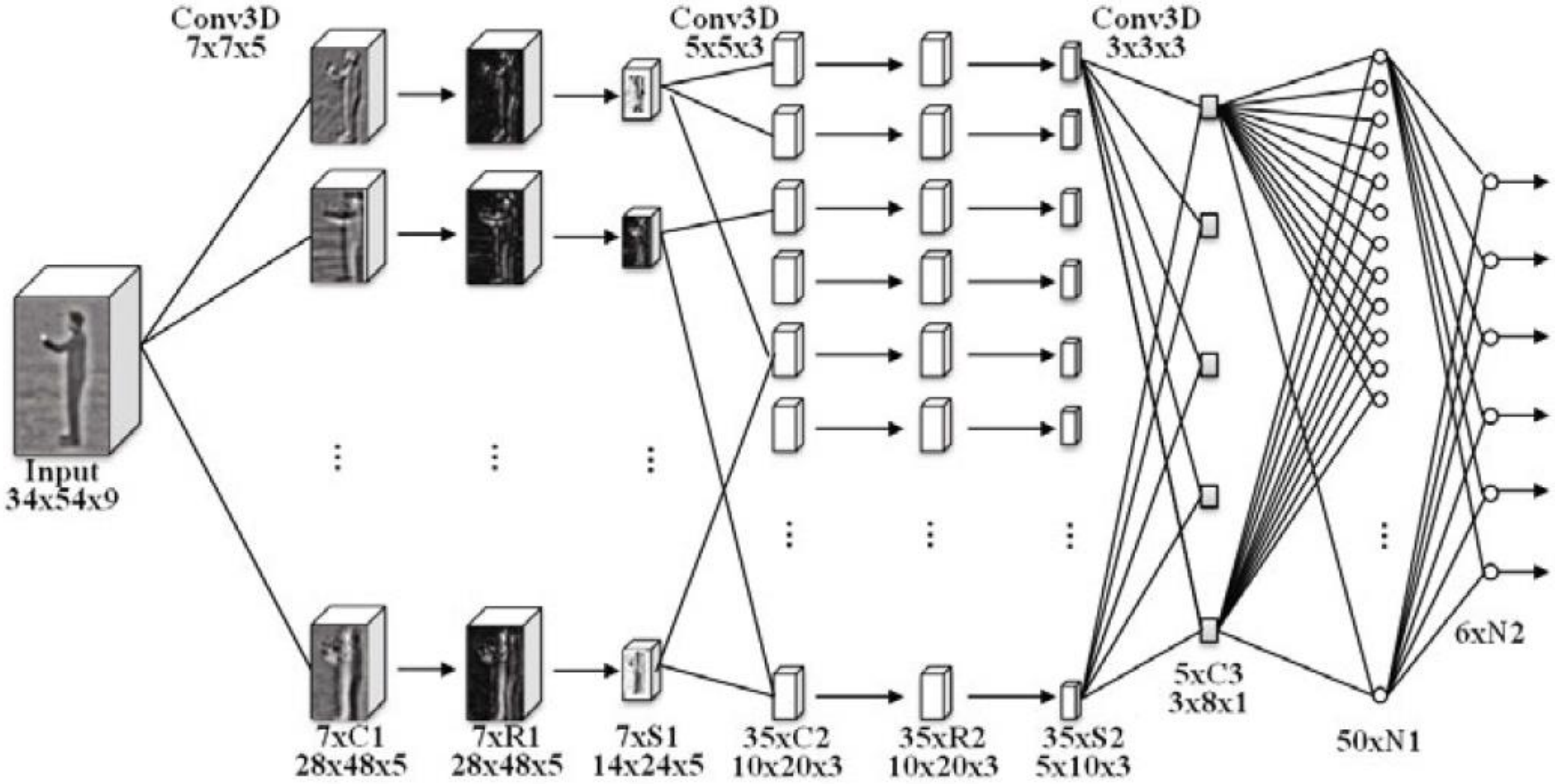


0_2	0_0	0_1	0	0	0	0
0_1	2_0	2_0	3	3	3	0
0_0	0_1	1_1	3	0	3	0
0	2	3	0	1	3	0
0	3	3	2	1	2	0
0	3	3	0	2	3	0
0	0	0	0	0	0	0

1	6	5
7	10	9
7	10	8

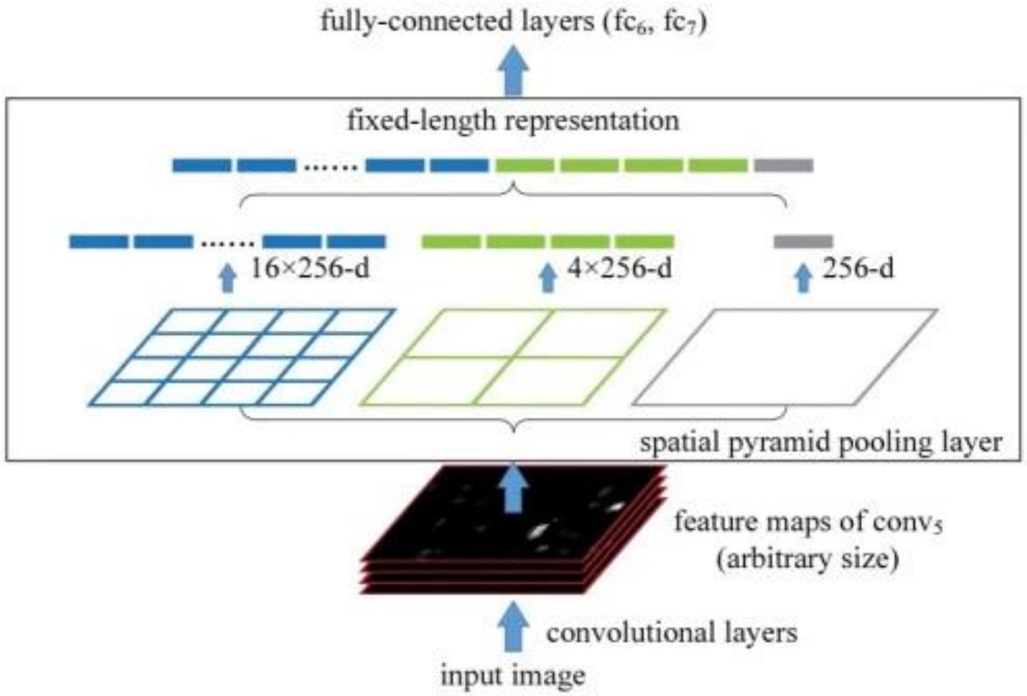


3D CNN for Action Recognition



Spatial Pyramid Pooling Layer

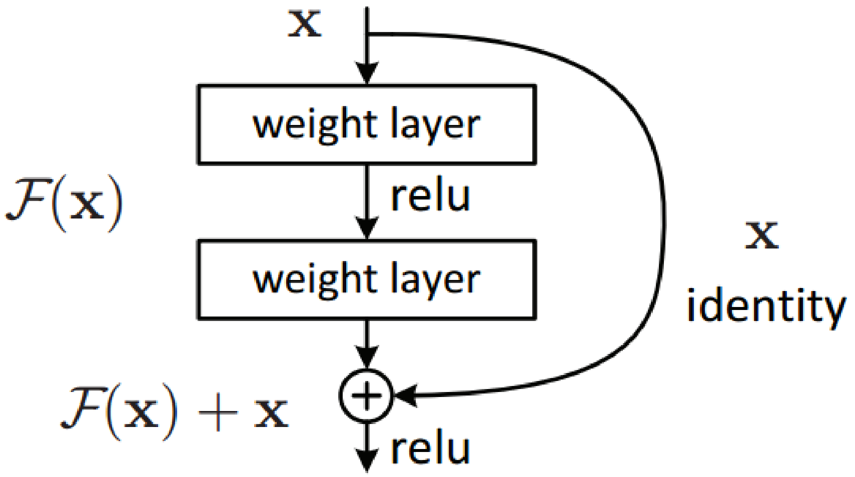
- **Spatial Pyramid Pooling** (SPP) allows CNN to use input images of any size, not only 224*224
- Convolutional layers operate on any size, but fully connected layers need fixed-size inputs
- Add a new SPP layer on top of the last convolutional layer, before the fully connected layer
- The SPP layer operates on each feature map independently.
- The output of the SPP layer is of dimension k*M, where k is the number of feature maps the SPP layer got as input and M is the number of bins
- Highly used in image segmentation (such as **DeepLab**) where image dimension has to be preserved



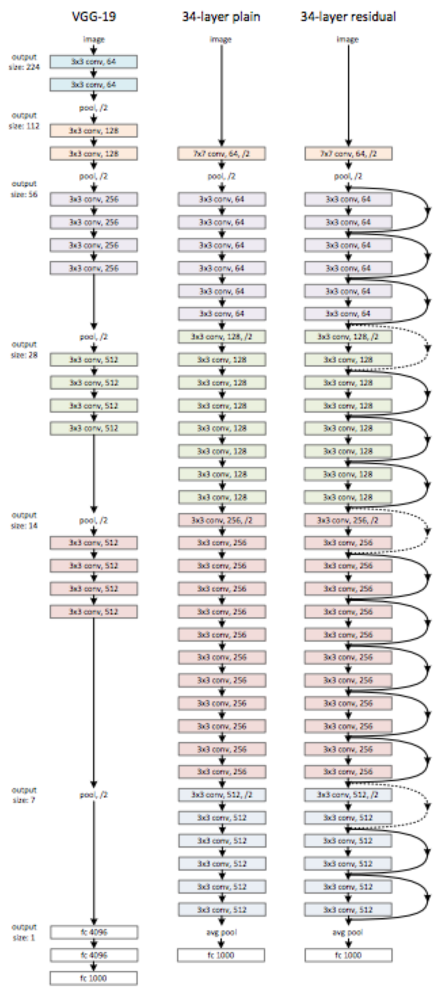
fully-connected layers (fc_6, fc_7)
fixed-length representation
16×256-d
4×256-d
256-d
spatial pyramid pooling layer
feature maps of $conv_5$
(arbitrary size)
convolutional layers
input image

Residual Block

- Deep Networks suffer from Vanishing Gradients
- Residual block allows us to add input with the output of successive layers to maintain the activation till the end
- **Constraints:**
  - In case of dense layers, number of nodes has to be same for i/p and o/p
  - In case of convolution, filters should be same for i/p and o/p. Padding is mandatory. Pooling should be done before applying the residual block
- Applied to all other domains of deep learning including speech and natural language processing



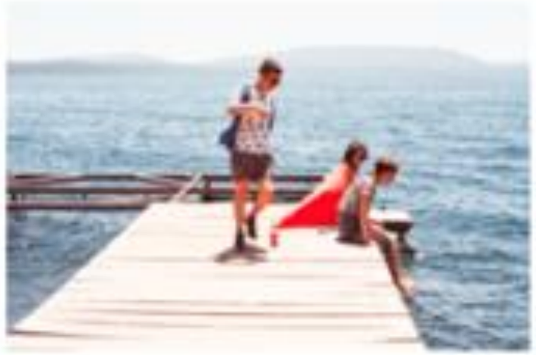
Residual Block



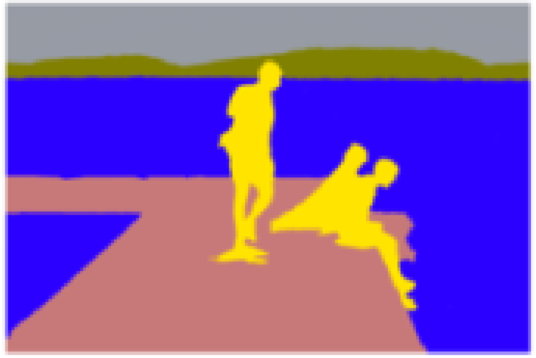
VGG16 vs ResNet

Semantic Segmentation

- *Semantic segmentation* describes the process of associating each pixel of an image with a class label such as *person*, *road*, *sky*, or *car*
- Traditional approaches involves unsupervised methods such as clustering, graph segmentation, etc.
- Unsupervised segmention is faster but fails to aggregate high-level visual features.
- Applications for semantic segmentation include:
  - Autonomous driving
  - Industrial inspection
  - Classification of terrain visible in satellite imagery
  - Medical imaging analysis



Input:
3 x H x W

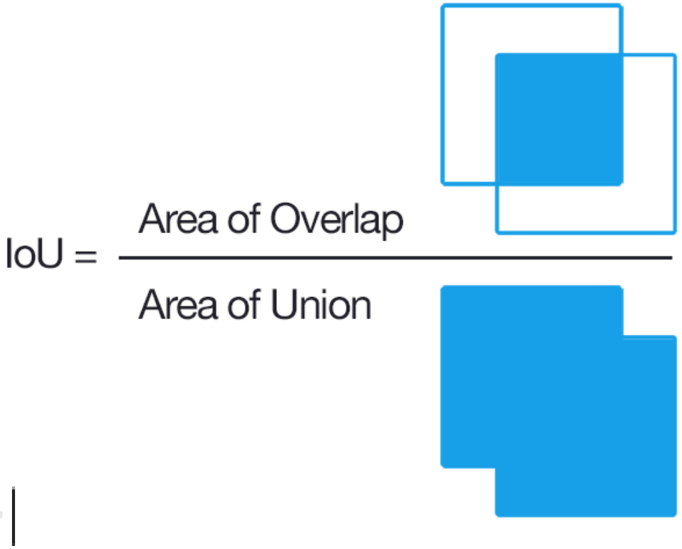


Predictions:
H x W

Segmentation Evaluation

- **Loss** for this network is computed by averaging the cross-entropy loss of every pixel and mini-batch
- **Accuracy**

$$acc(P, GT) = \frac{|\text{pixels correctly predicted}|}{|\text{total nb of pixels}|}$$

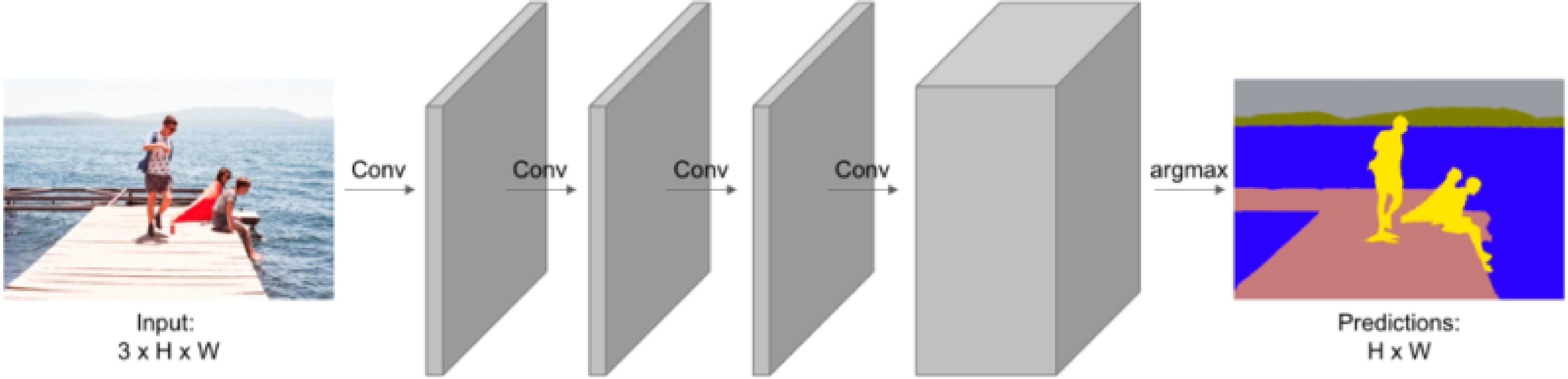


$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

- **Jaccard (Intersection over Union)**

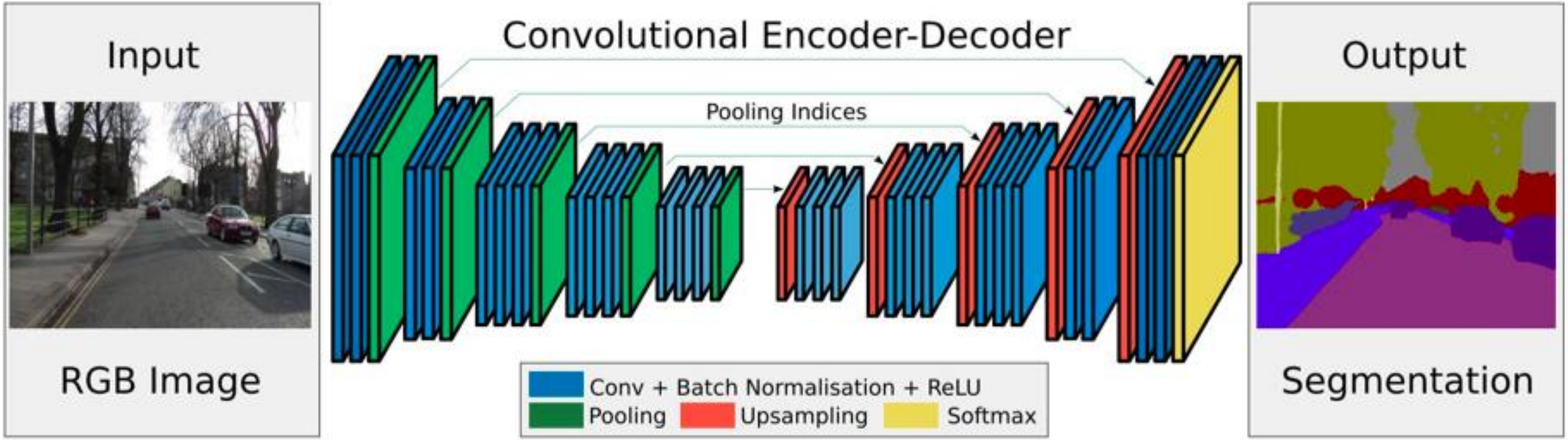
$$jacc(P(class), GT(class)) = \frac{|P(class) \cap GT(class)|}{|P(class) \cup GT(class)|}$$

Fully Convolutional Network (FCN)

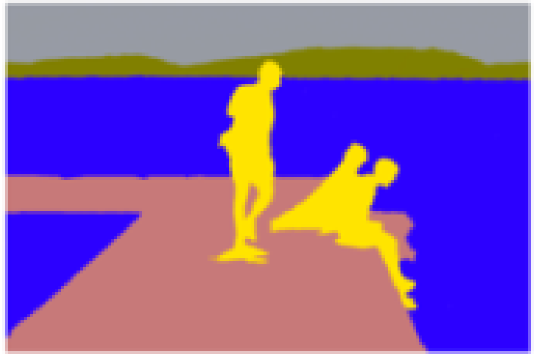


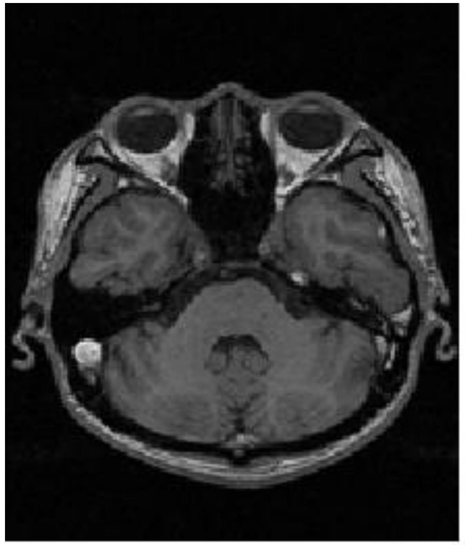
- The final layer has a depth equal to the number of classes. FCN is similar to object detection except that the spatial dimension is preserved.
- The output produced by the architecture will be coarse as some pixels may be mis-predicted, while the computation is high.

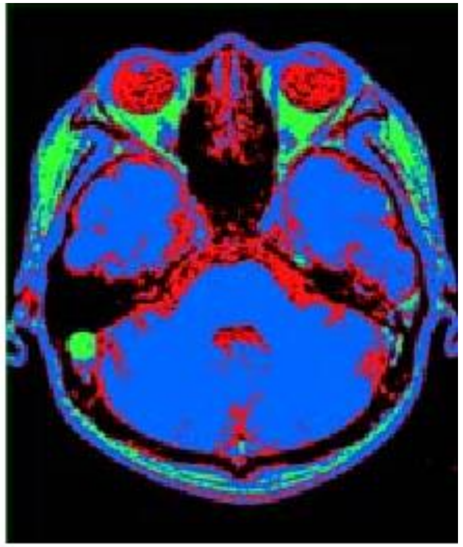
SegNet

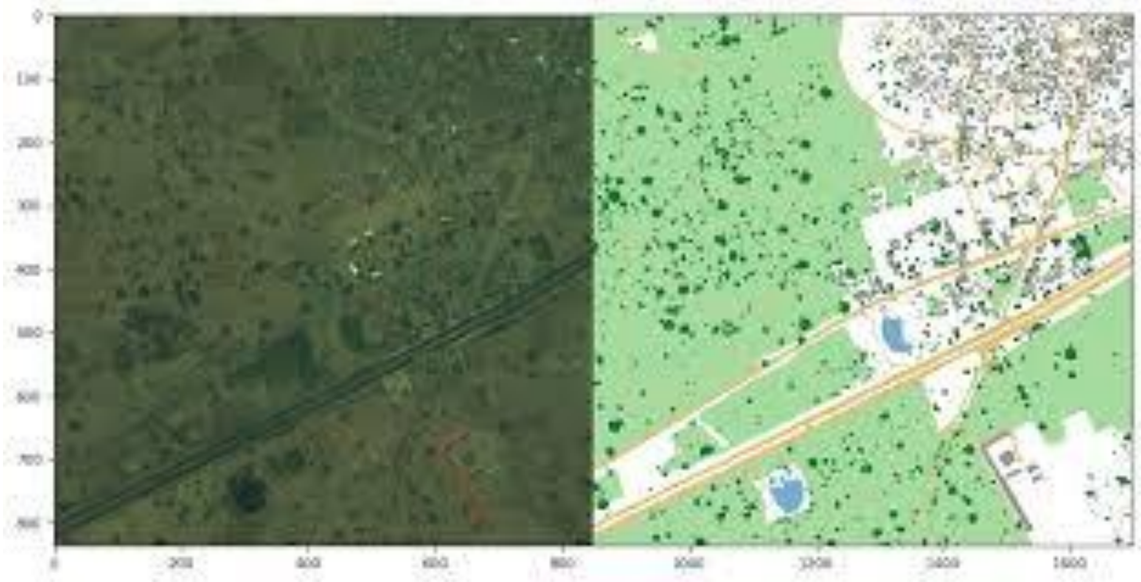

Input
RGB Image
Convolutional Encoder-Decoder
Pooling Indices
Output
Segmentation
Conv + Batch Normalisation + ReLU
Pooling
Upsampling
Softmax

Medical/Satellite Image Segmentation



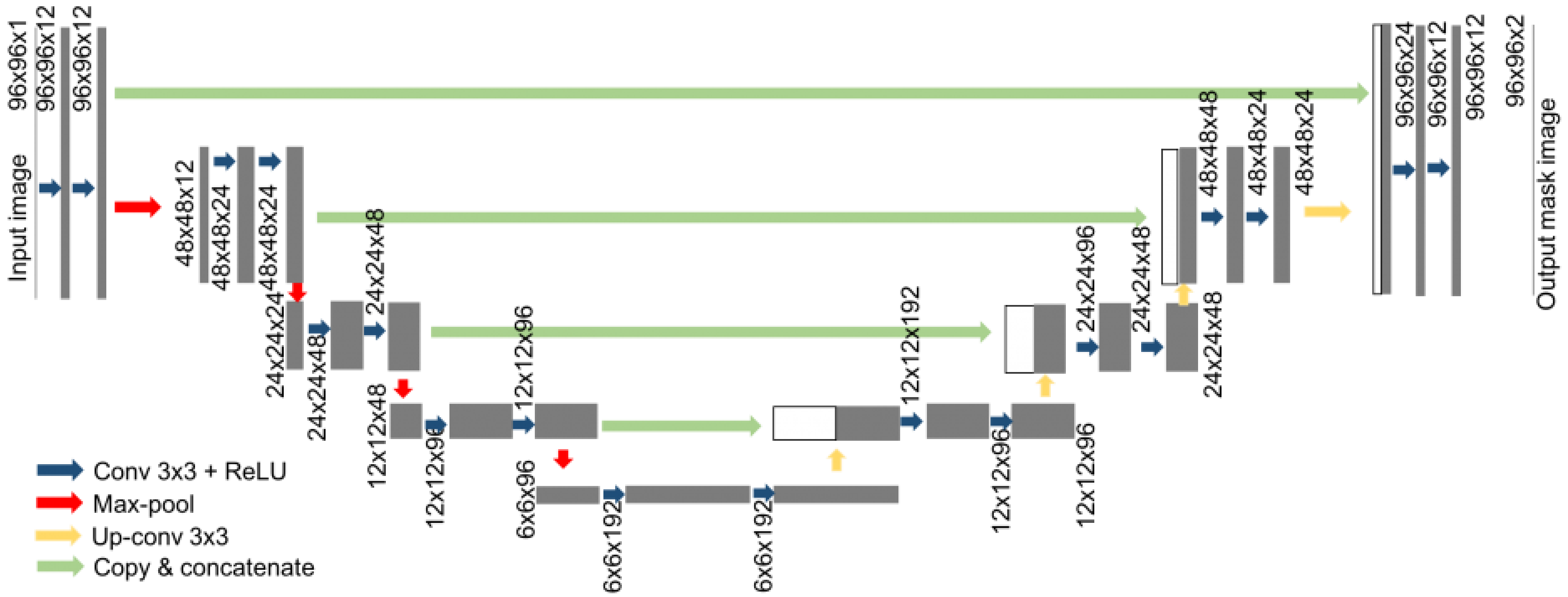




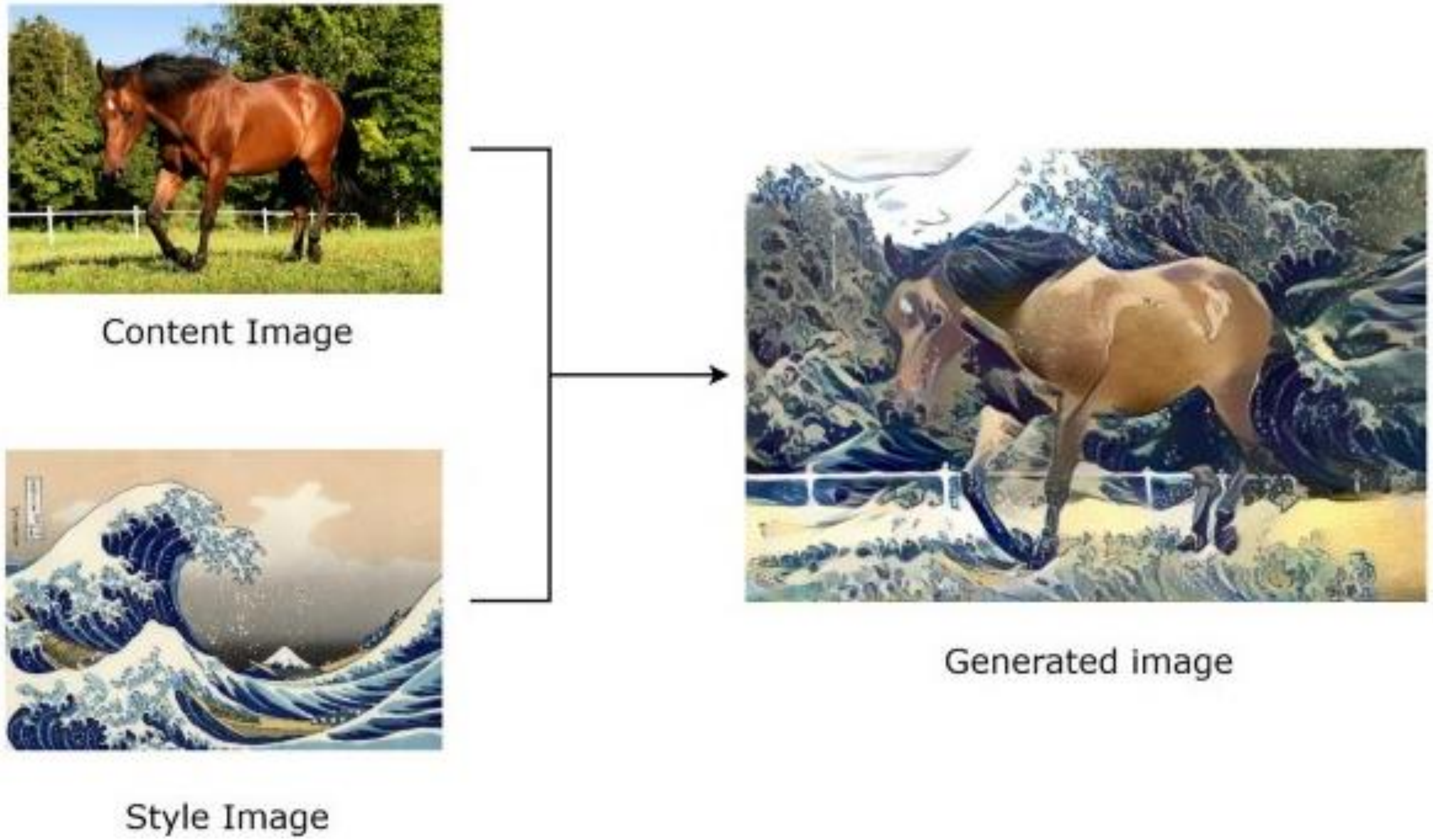


UNet Architecture

- UNet is a symmetric encoder-decoder network. Highly used in satellite/medical image segmentation
- Encoder: Contraction path, Decoder: Expansion path. Feature detection and location is used at decoder part



Neural Style Transfer



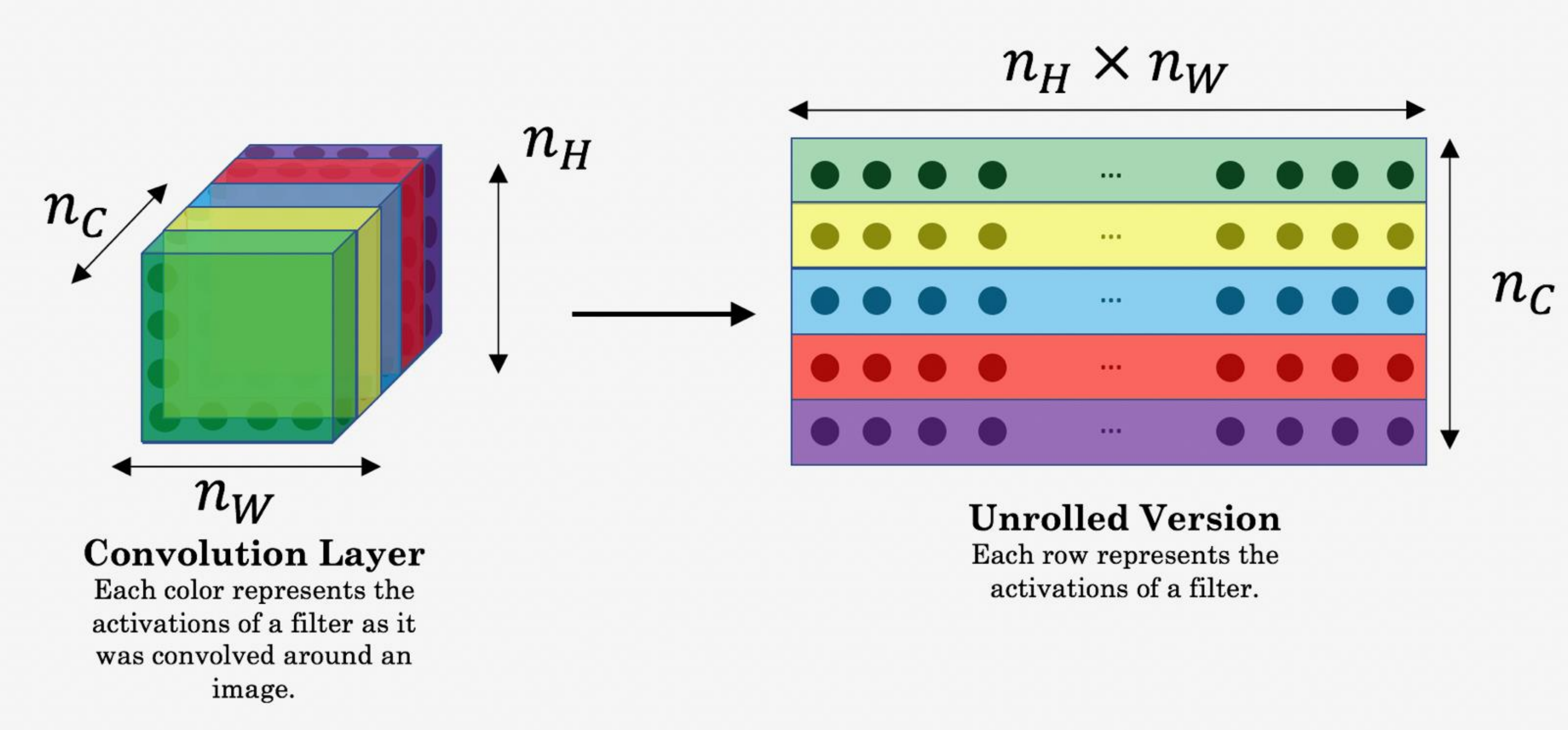
Content Image

Style Image

Generated image

- Cost Function $J(G) = \alpha J_{content}(C, G) + \beta J_{style}(S, G)$

Neural Style Transfer



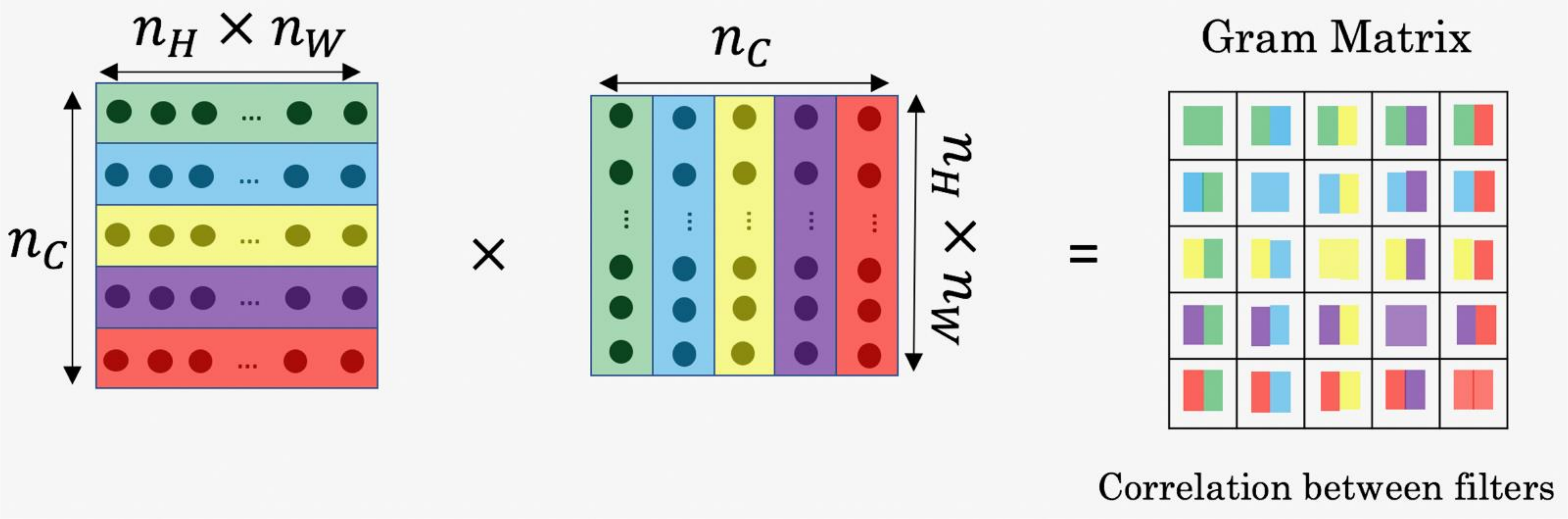
Convolution Layer
Each color represents the activations of a filter as it was convolved around an image.

Unrolled Version
Each row represents the activations of a filter.

- Content Cost Function

$$J_{content}(C,G) = \frac{1}{4 \times n_H \times n_W \times n_C} \sum_{\text{all entries}} (a^{(C)} - a^{(G)})^2$$

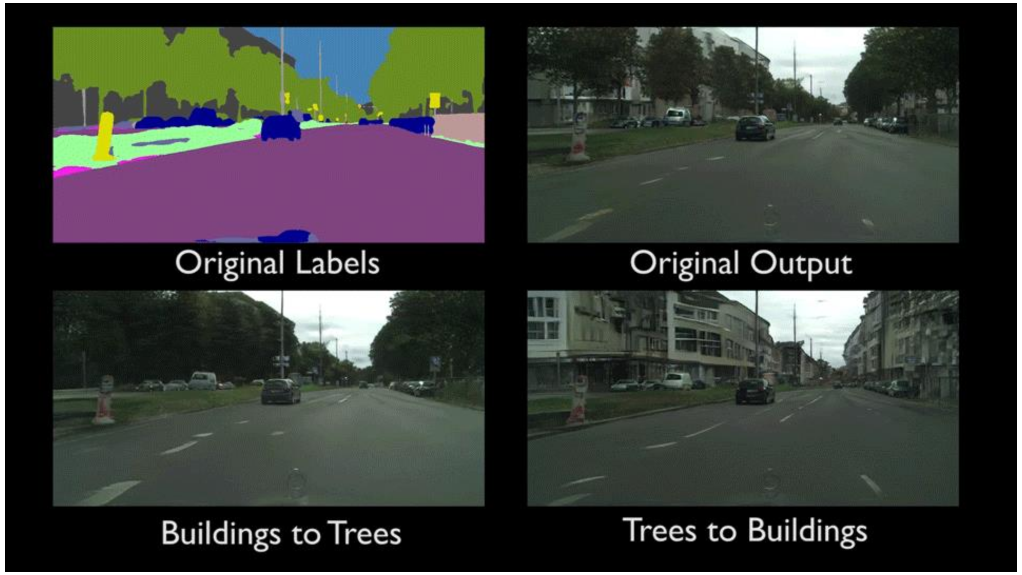
Neural Style Transfer

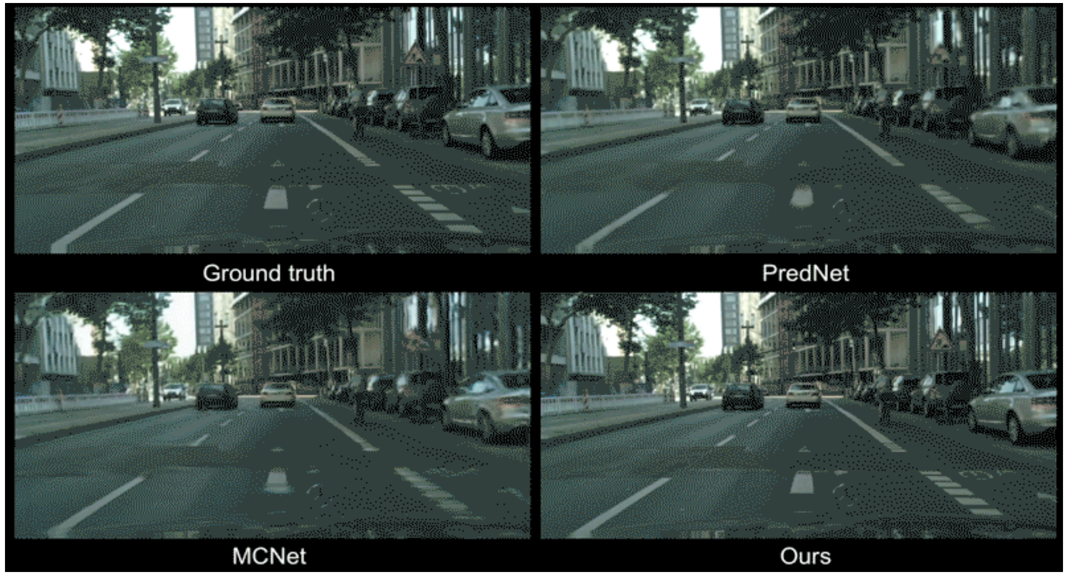


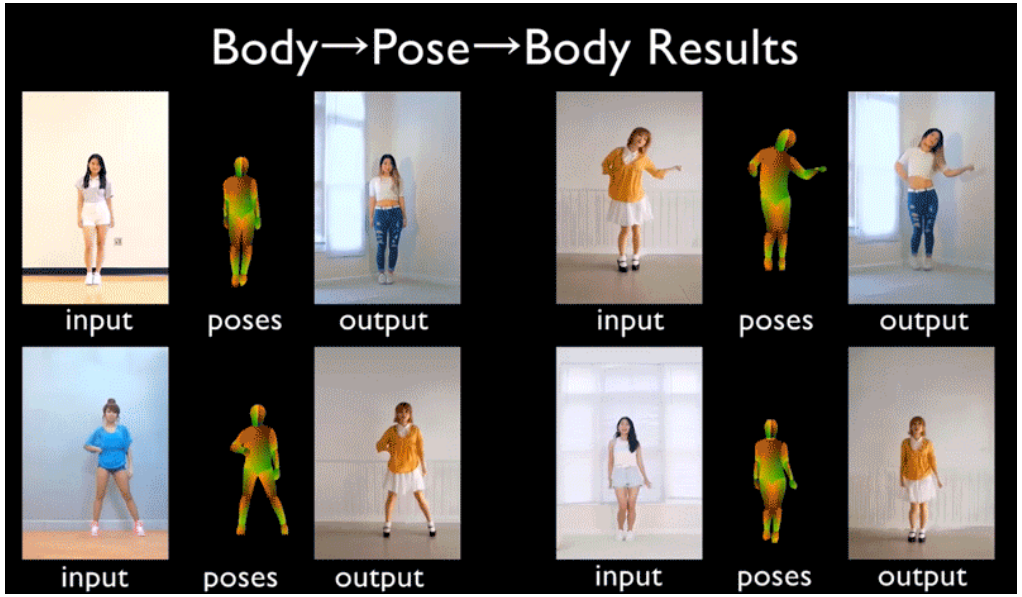
- Style Cost Function

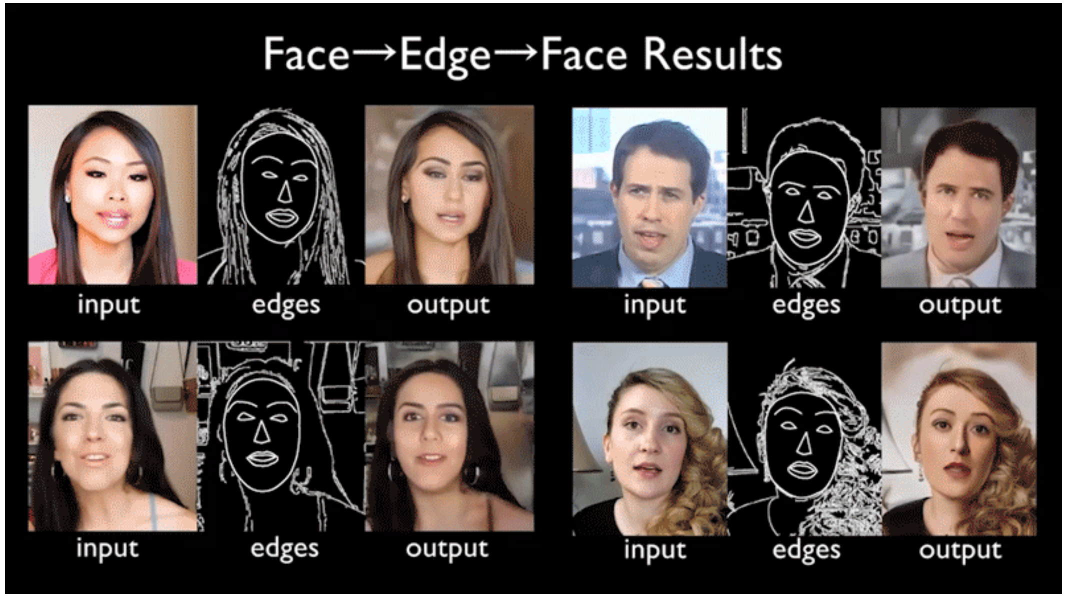
$$J_{style}^{[l]}(S,G) = \frac{1}{4 \times {n_C}^2 \times (n_H \times n_W)^2} \sum_{i=1}^{n_C} \sum_{j=1}^{n_C} (G_{ij}^{(S)} - G_{ij}^{(G)})^2$$

Video to Video Synthesis



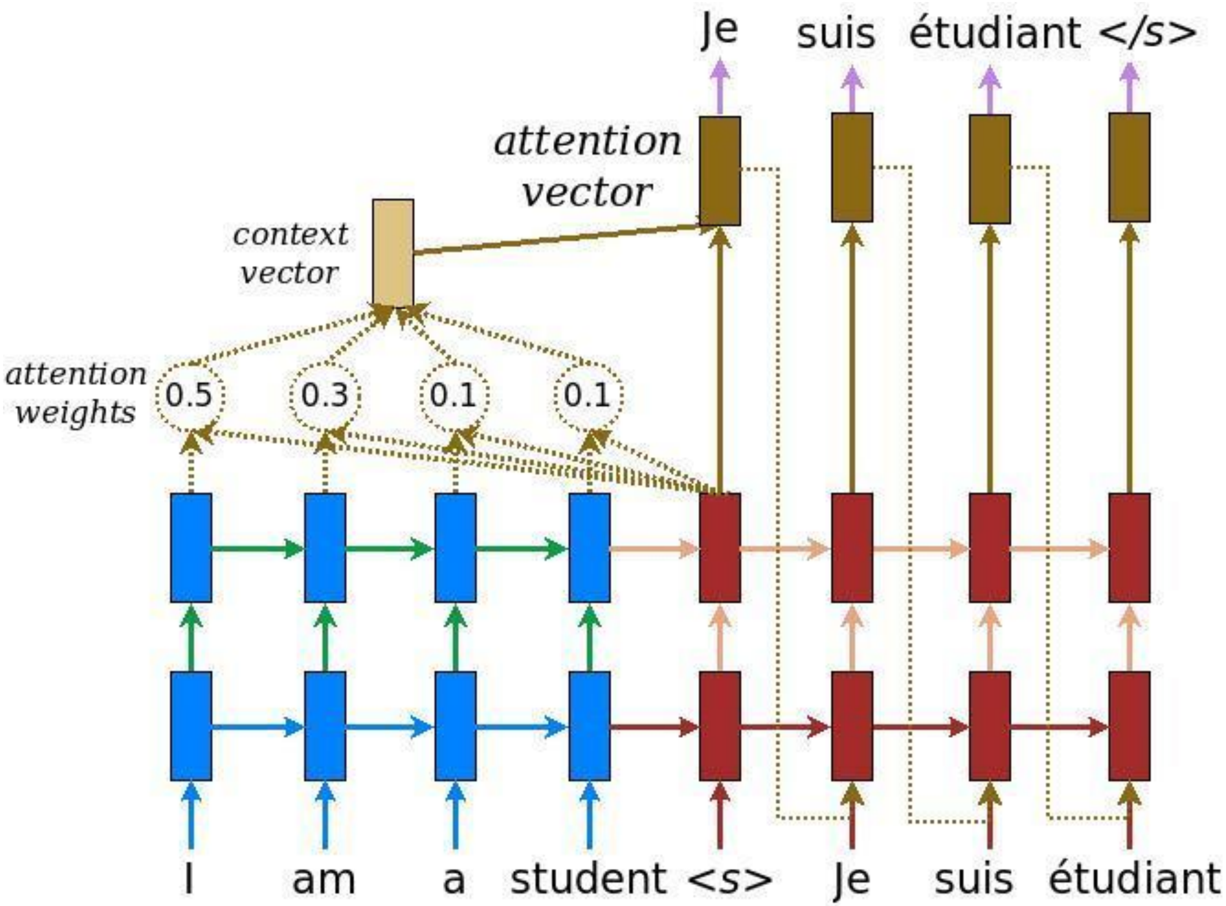


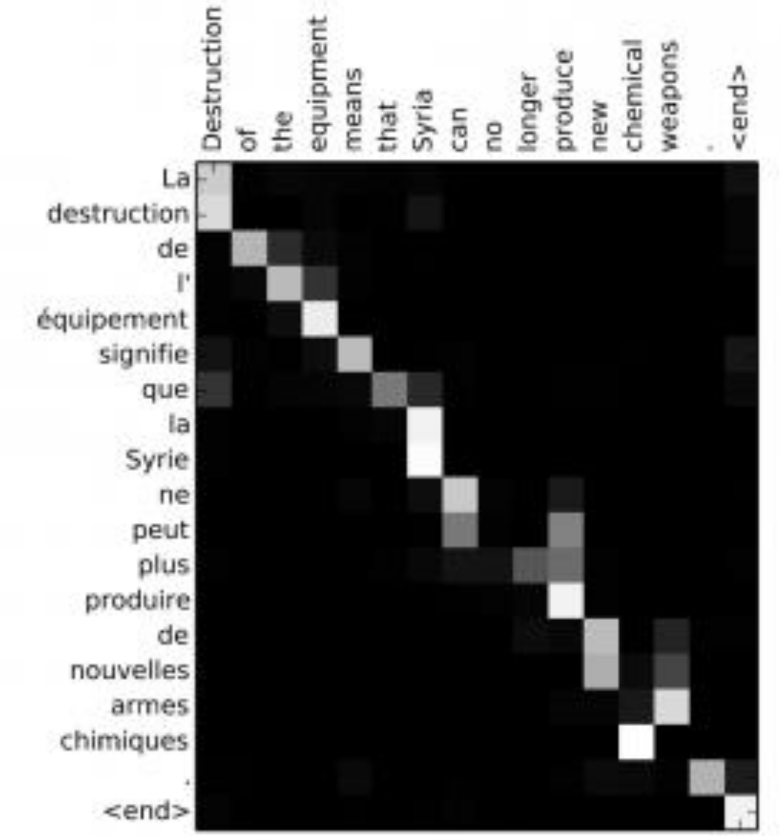




Attention Model

- Attention mechanism helps us to focus on specific parts of input to generate specific output.

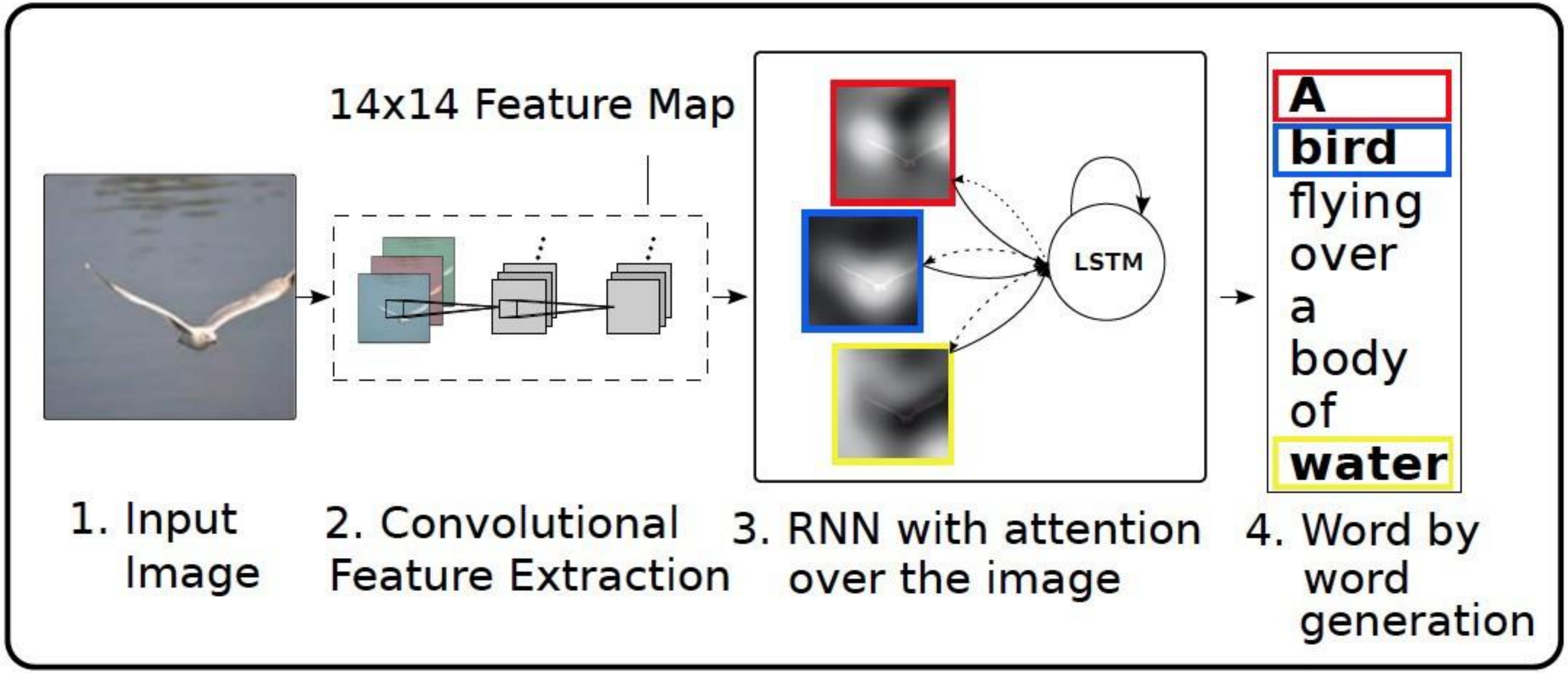




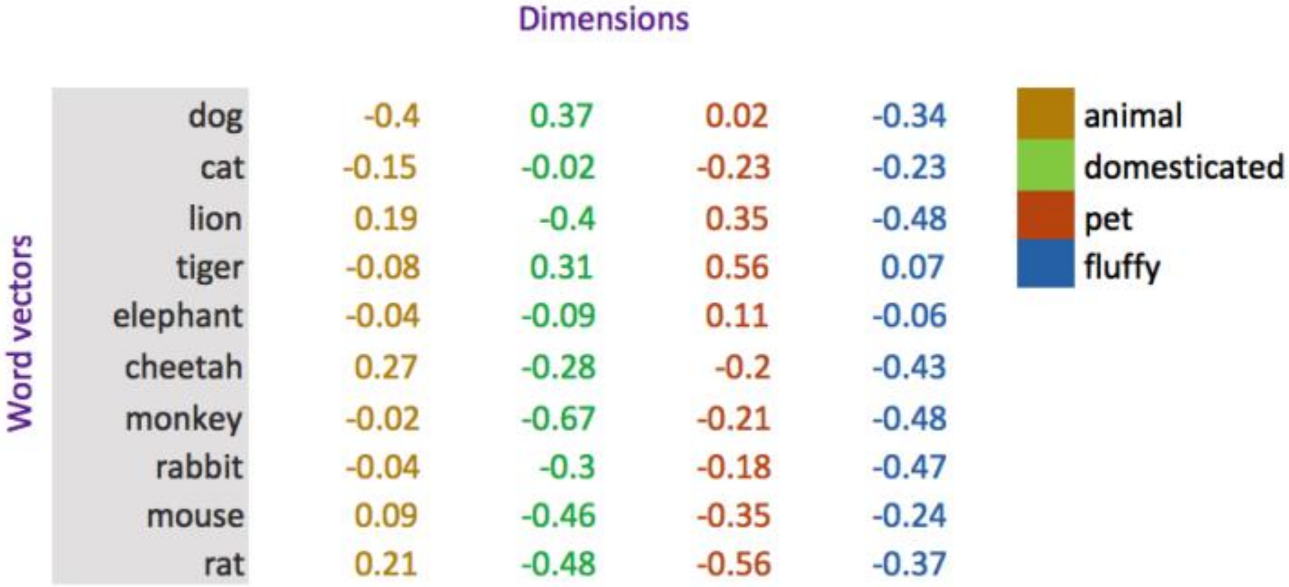
Attention Weight Matrix

Attention Model for Image Captioning

- Attention model in images helps to focus on specific parts of image


14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

Word Embeddings

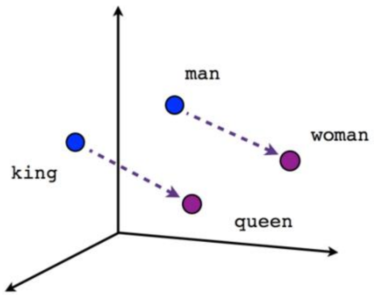


Dimensions

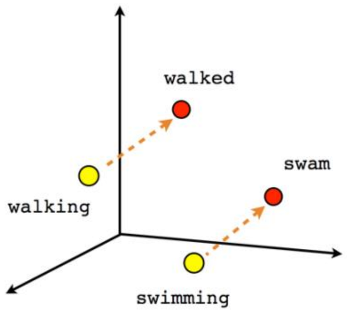
Word vectors

dog	-0.4	0.37	0.02	-0.34
cat	-0.15	-0.02	-0.23	-0.23
lion	0.19	-0.4	0.35	-0.48
tiger	-0.08	0.31	0.56	0.07
elephant	-0.04	-0.09	0.11	-0.06
cheetah	0.27	-0.28	-0.2	-0.43
monkey	-0.02	-0.67	-0.21	-0.48
rabbit	-0.04	-0.3	-0.18	-0.47
mouse	0.09	-0.46	-0.35	-0.24
rat	0.21	-0.48	-0.56	-0.37

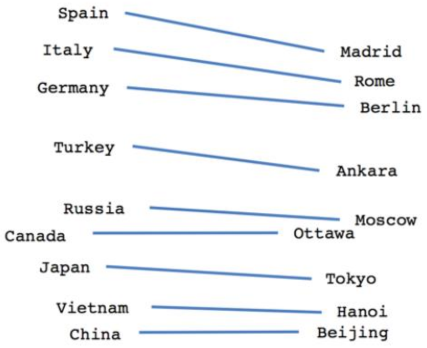
- 🟫 animal
- 🟩 domesticated
- 🟥 pet
- 🟦 fluffy



Male-Female

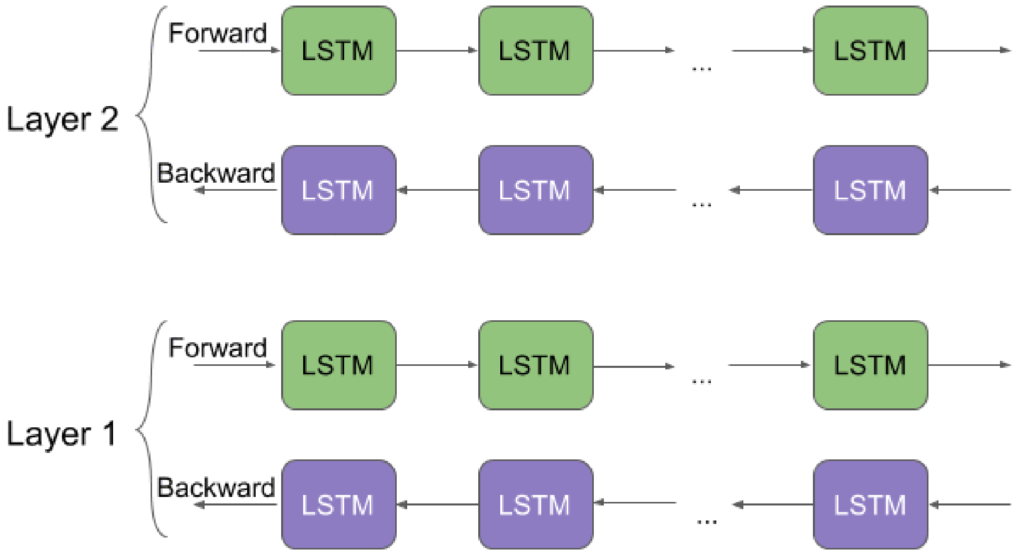


Verb tense



Country-Capital

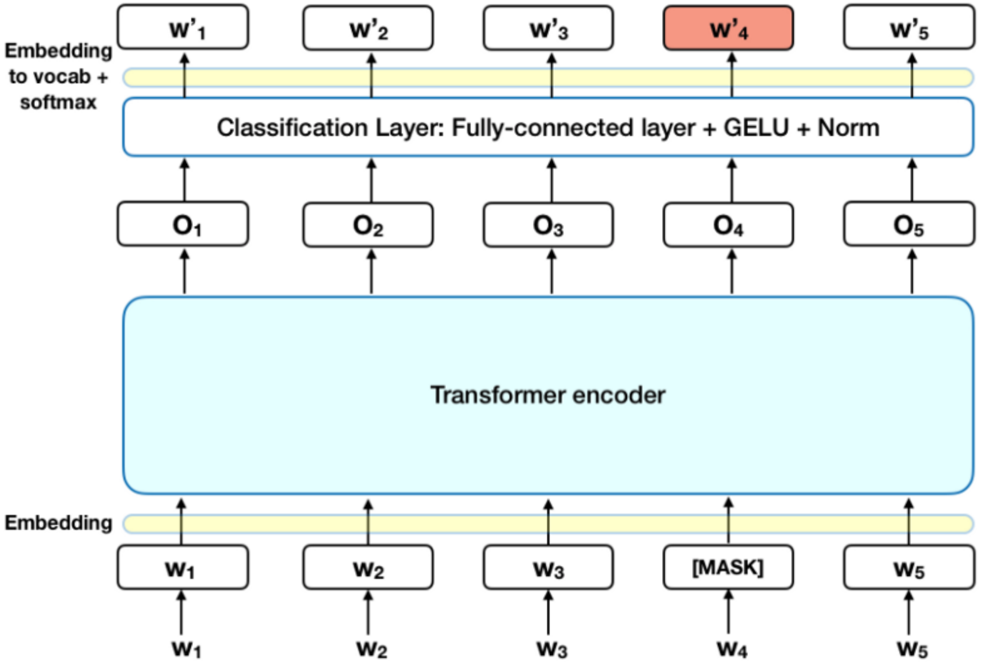
ELMo



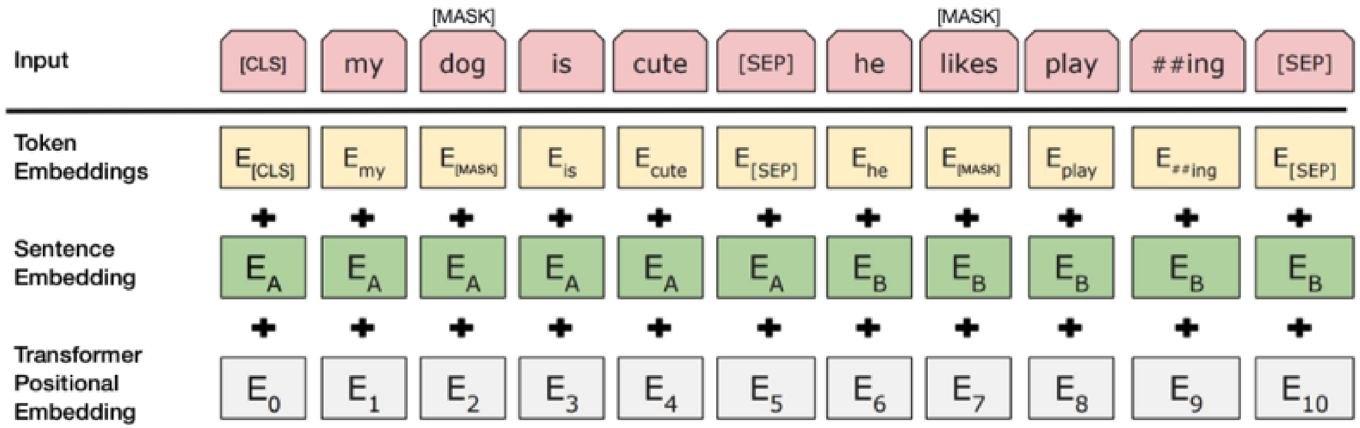
- ELMo (Embeddings from Language Models) is a bidirectional character-level language mode for Word Embedding
- ELMo helps us to handle **Polysemy** wherein a word could have multiple meanings or senses.
  - I ***read*** the book yesterday.
  - Can you ***read*** the letter now?

BERT

- BERT - Bidirectional Encoder Representations from Transformers
- Model is good at generating context based embeddings



Masked LM (MLM)



Next Sentence Prediction (NSP)

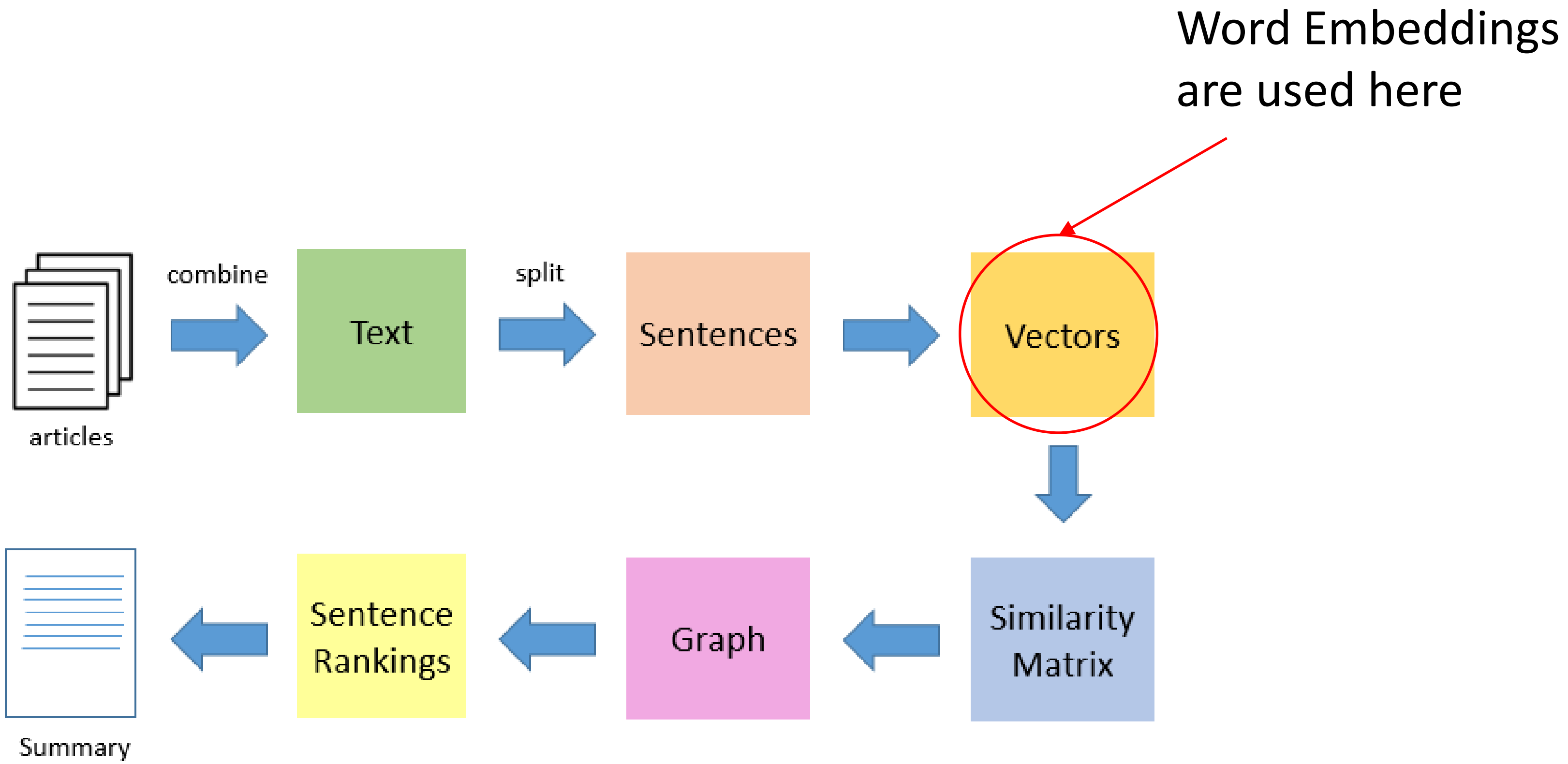
MUSE: Multilingual Unsupervised and Supervised Embeddings

- MUSE is a Python library for *multilingual word embeddings*
- Provides state-of-the-art multilingual word embeddings
- Large-scale high-quality bilingual dictionaries for training and evaluation
- https://github.com/facebookresearch/MUSE

Pretrained NLP Models

- **Multi-Purpose NLP Models**
  - ULMFiT
  - Transformer
  - Google's BERT
  - Transformer-XL
  - OpenAI's GPT-2
- **Word Embeddings**
  - ELMo
  - Flair
- **Other Pretrained Models**
  - StanfordNLP

TextRank Algorithm - Summarization



Questions ?

Thanks