SHIP CLASSIFICATION & SEGMENTATION

Nazih Kalo

OUTLINE

1. Introduction (Objective, Dataset, Recap Computer Vision)
2. EDA
3. Image Classification
4. Semantic Segmentation
5. Conclusion

INTRODUCTION
(OBJECTIVE, DATASET, RECAP COMPUTER VISION)

OBJECTIVE

"Locate ships in images, and put an aligned bounding box segment."

Notes:

- Images may contain no ships, single ship or many ships
- Ships can be located in open sea, at docks, marinas, etc.

DATASET

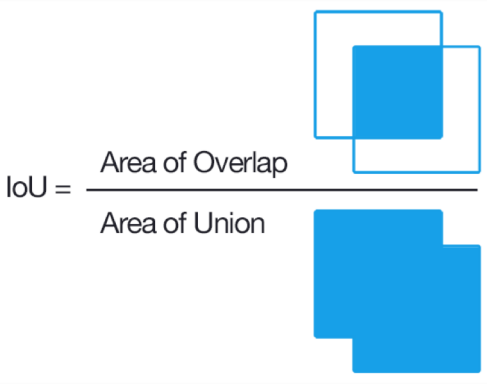
Training & Test Sets:

- *train.zip:*
  - 192,556 images
  - 26.4GB
- *test.zip*
  - 15,606 images
  - 2.2 GB
- *train_ship_segmentations.csv:*
  - Provides the ground truth for the training images
  - Masks are in run-length encoding format (e.g. '1 3' implies starting at pixel 1 and running a total of 3 pixels (1,2,3))

	ImageId	EncodedPixels
0	00003e153.jpg	NaN
1	0001124c7.jpg	NaN
2	000155de5.jpg	264661 17 265429 33 266197 33 266965 33 267733...
3	000194a2d.jpg	360486 1 361252 4 362019 5 362785 8 363552 10 ...
4	000194a2d.jpg	51834 9 52602 9 53370 9 54138 9 54906 9 55674 ...

Evaluation Metric:

- F2 Score at different intersection over union (IoU) thresholds



$$F_2 = 5 \cdot \frac{precision \cdot recall}{4 \cdot precision + recall}$$

- F2 Score is the weighted average of precision and recall
- Weights recall higher than precision

RECAP: COMPUTER VISION

Image Classification

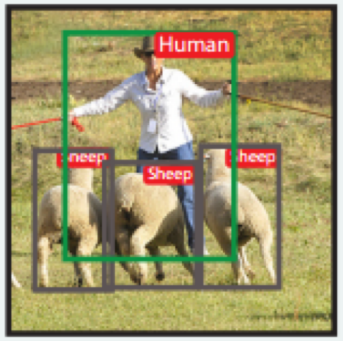
Classify an image based on the dominant object inside it.

datasets: MNIST, CIFAR, ImageNet

Object Localization

Predict the image region that contains the dominant object. Then image classification can be used to recognize object in the region

datasets: ImageNet



Object Recognition

Localize and classify all objects appearing in the image. This task typically includes: proposing regions then classify the object inside them.

datasets: PASCAL, COCO

Semantic Segmentation

Label each pixel of an image by the object class that it belongs to, such as human, sheep, and grass in the example.

datasets: PASCAL, COCO

Instance Segmentation

Label each pixel of an image by the object class and object instance that it belongs to.

datasets: PASCAL, COCO

Keypoint Detection

Detect locations of a set of predefined keypoints of an object, such as keypoints in a human body, or a human face.

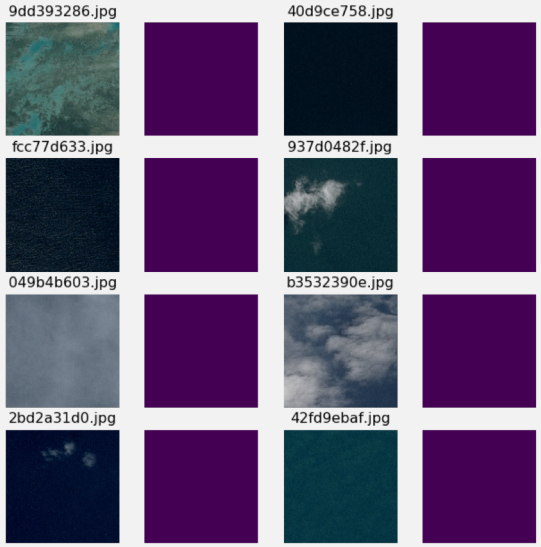
datasets: COCO

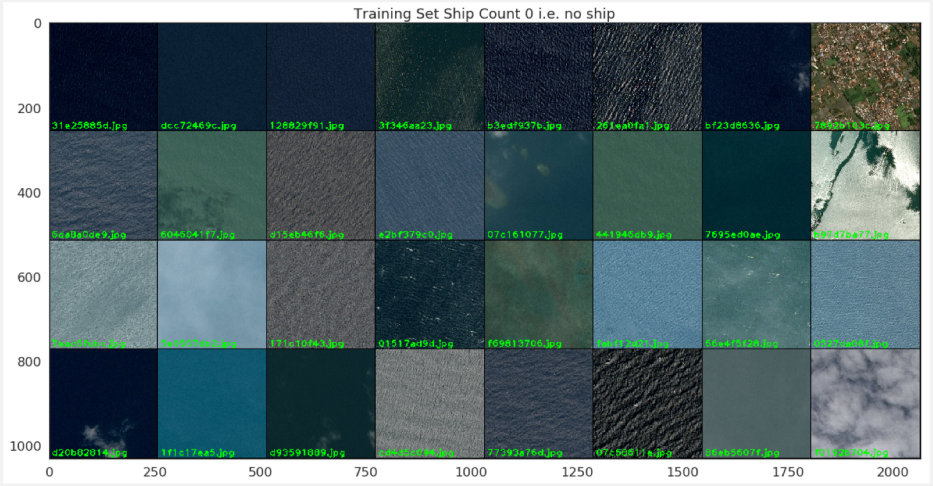
- Our modeling & analysis will be limited to Image Classification
- We will also explain the modeling techniques involved in Semantic Segmentation (since it is the objective of the Kaggle Competition)

EXPLORATORY DATA ANALYSIS

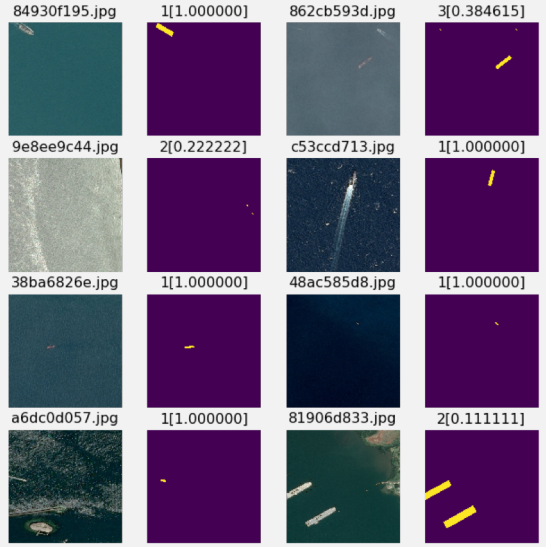
EDA

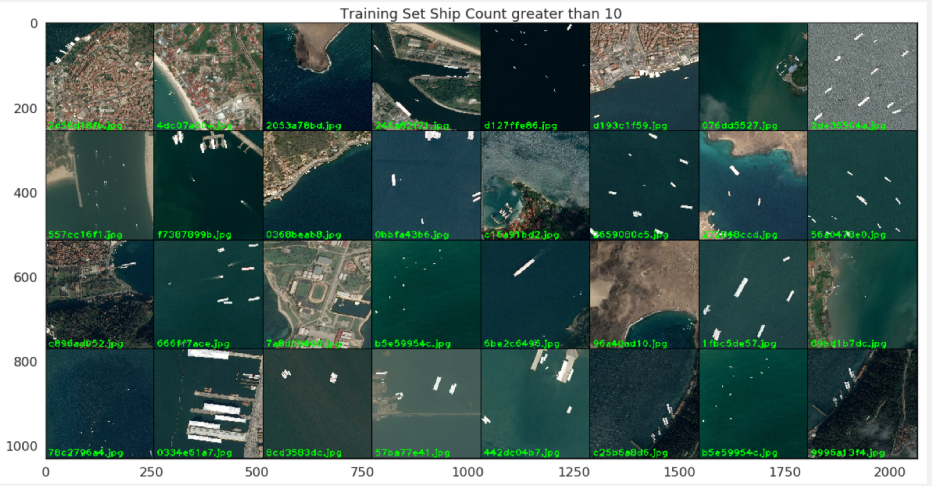
Images with no ships along with their masks:





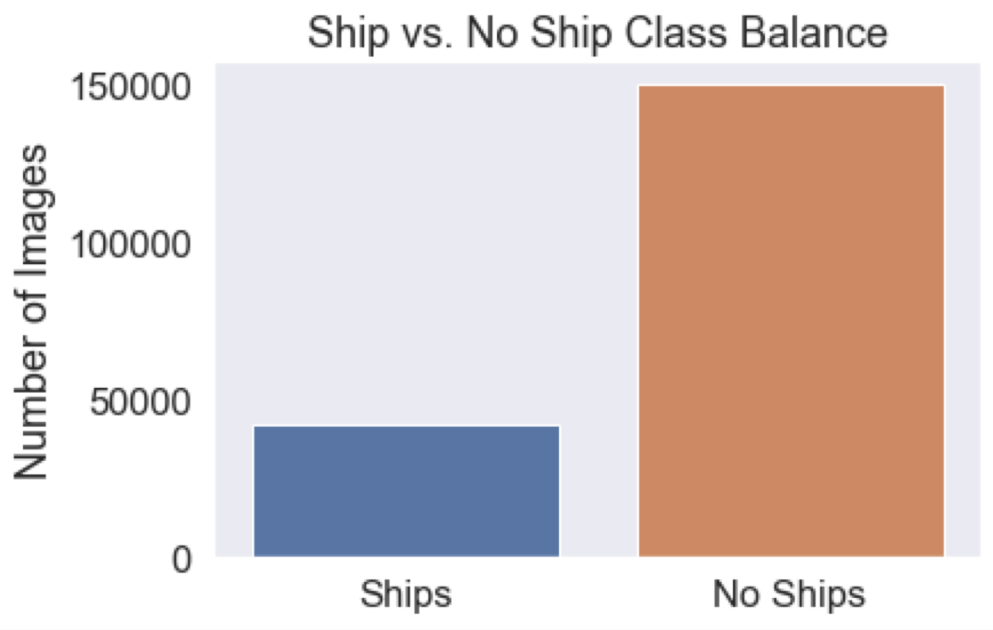
Images with ships along with their masks:





EDA

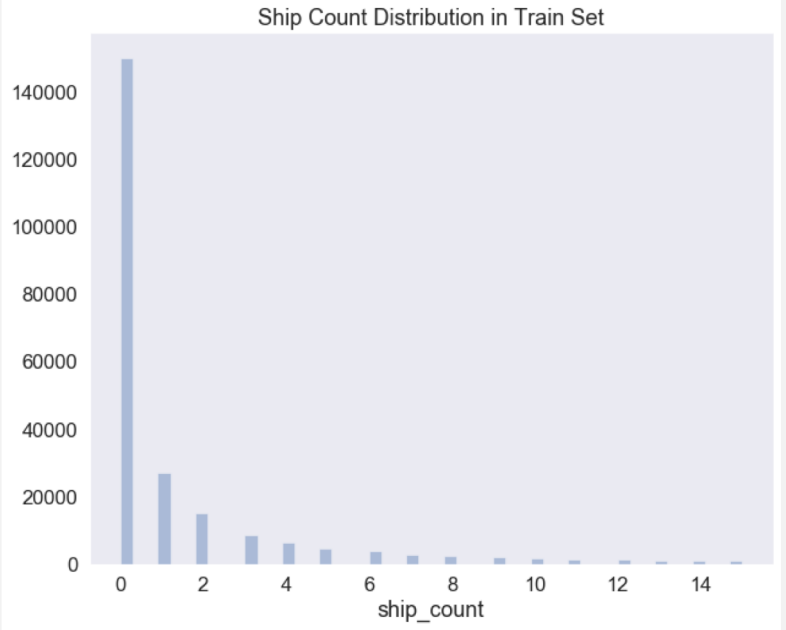
Ship vs. No Ship:



- Dataset is highly imbalanced
- ~ 3x # of images without ships than with ships

Need to balance!

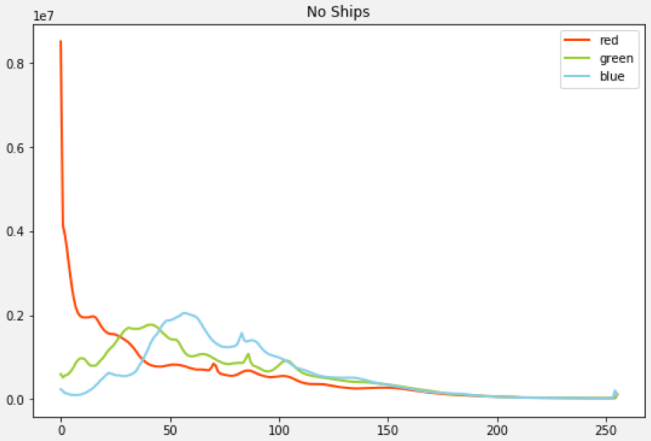
Distribution of # of ships per image:

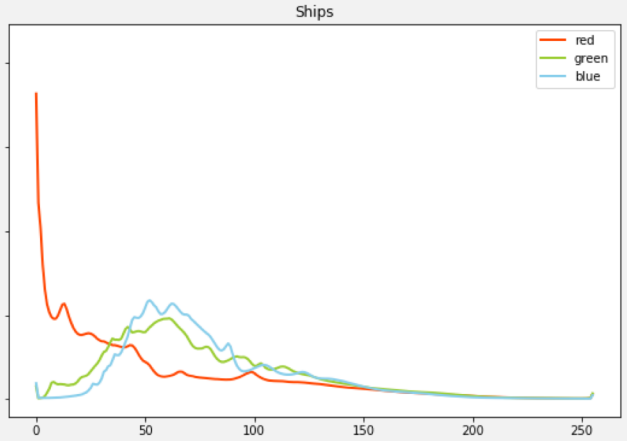


- Most images that have a ship in it have between 1-4 ships
- Some extreme images with 10 or more ships

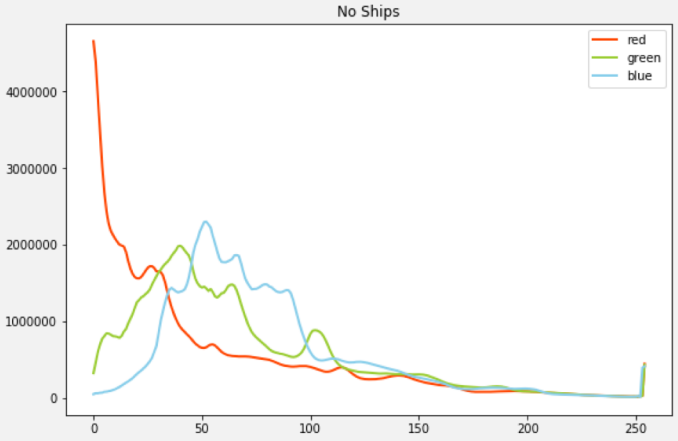
EDA

Look at color distributions between images with ships and those without:





Look at color distributions of areas with no ships and ships themselves:



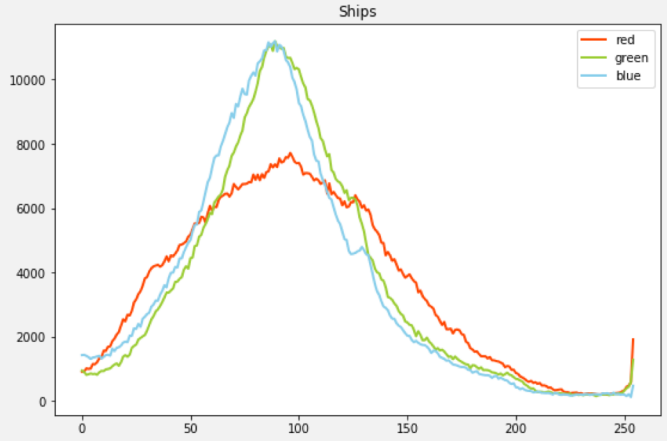


IMAGE CLASSIFICATION

IMAGE CLASSIFICATION

Using transfer learning (CNN) for binary image classification task

Overview:

- Rather than trying to segment, we build a model that tries to identify if any boat shows up in the image
- We are using transfer learning ('DenseNet169')
- Going on, the model could be useful as a quick way to screen through a lot of images to see if they are likely to have a boat and if they are then run a more expensive full-resolution model on that sample

---

Motivation of using CNN:

- CNNs are very good feature extractors that learn filters/characteristics (extracting useful attributes from an already trained CNN with its trained weights)
- Reduces images to a form which is easier to process, without losing relevant features
- Less pre-processing required
- Can provide a reasonable baseline result based on which further improvements can be incorporated

IMAGE CLASSIFICATION

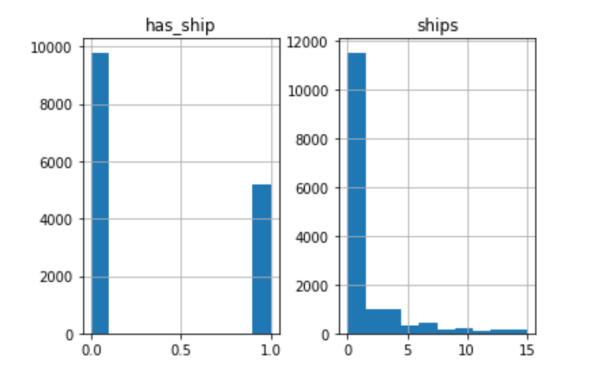
Data Processing & Augmentation

Processing:

- Convert 'EncodedPixels' column to binary outcome

	ImageId	EncodedPixels	ship_bool
0	00003e153.jpg	NaN	0
1	0001124c7.jpg	NaN	0
2	000155de5.jpg	264661 17 265429 33 266197 33 266965 33 267733...	1
3	000194a2d.jpg	360486 1 361252 4 362019 5 362785 8 363552 10 ...	1
4	000194a2d.jpg	51834 9 52602 9 53370 9 54138 9 54906 9 55674 ...	1

- Balanced by down-sampling dataset to 15,000 obs
- Stratified train-test-split (70/30) by 'ship_bool'



Sources:
[1] Ho et al. (2019): https://bair.berkeley.edu/blog/2019/06/07/data_aug/ .

Augmentation:

"Data augmentation is a strategy that enables practitioners to significantly increase the diversity of data available for training models, without actually collecting new data. Data augmentation techniques such as cropping, padding, and horizontal flipping are commonly used to train large neural networks." [1]

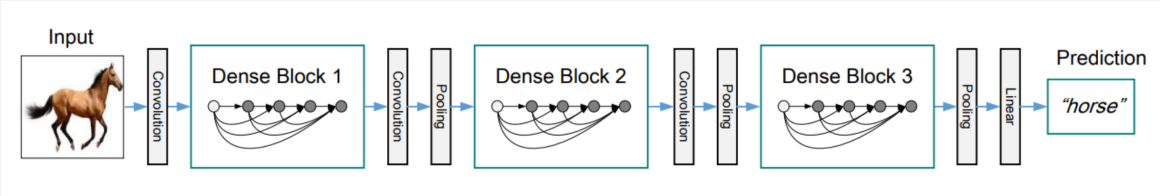
Augmentation techniques used:

- Rotation, Shifting, Adjusting Brightness, Zooming, Flipping
- Example of common techniques:

IMAGE CLASSIFICATION

Densely Connected Convolutional Networks [1]

Concept:



- Connects each layer to every other layer in a feed-forward fashion
- Has $\frac{L(L+1)}{2}$ direct connections for L layers (compared to L connections in traditional convolutional networks)
- feature-maps of all preceding layers are used as inputs, and its own feature-maps are used as inputs into all subsequent layers

Advantages:

- Alleviate the vanishing-gradient problem
- Strengthen feature propagation
- Substantially reduce the number of parameters
- More diversified features
- Well suited for various computer vision tasks built on convolutional features

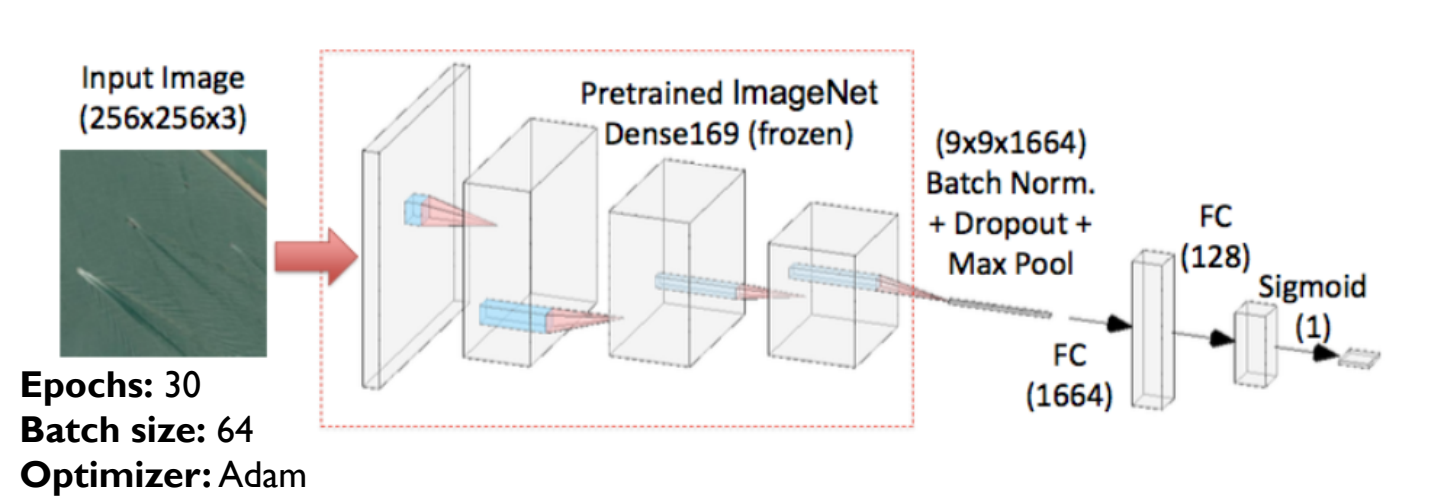
Disadvantages:

- Requires enormous amount of memory

Sources:
[1] Huang et al., (2016): Densely Connected Convolutional Networks.

IMAGE CLASSIFICATION

Pre-trained CNN approach [1]



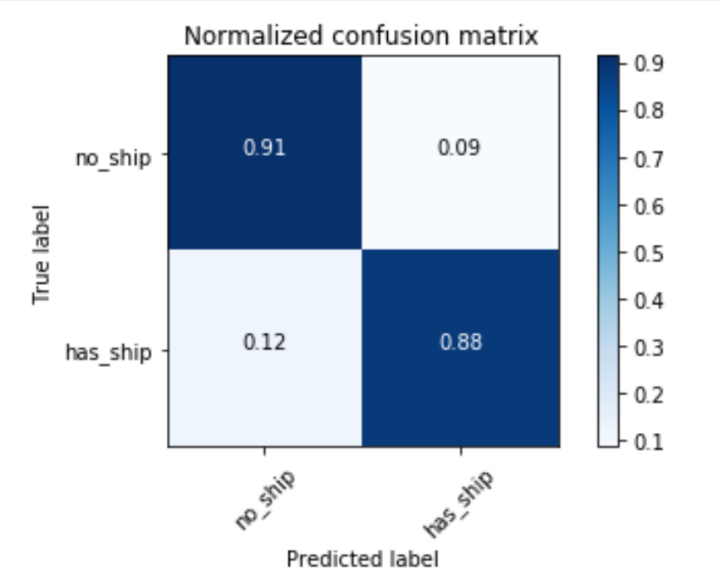
- Cross-entropy loss for loss function
- Batch Normalization is applied to improve training speed, enable increased learning rate, and reduce covariance shift (important, because 'DenseNet169' was not trained on ships)
- Dropout to reduce overfitting
- Max Pooling to prevent overfitting and reduce computational cost by reducing number of parameters
- Decaying learning rate to facilitate model training
- Weights of model would not be updated, if the loss of development set did not improve after training an epoch

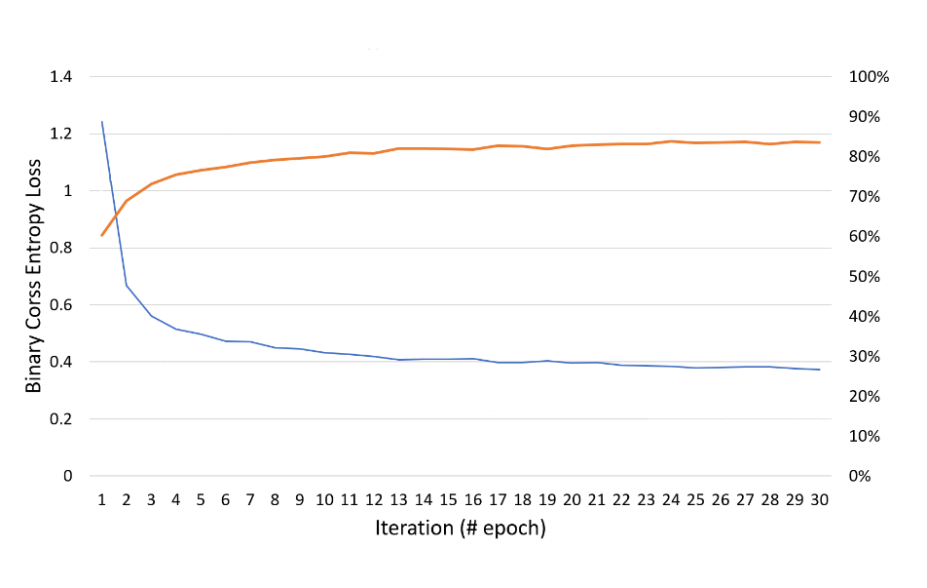
Sources:

[1] Chen et al., (2018): Traditional vs. Convolutional Neural Network Approach.

IMAGE CLASSIFICATION

Results





- **Precision:** 0.91
- **Recall:** 0.88
- **Specificity:** 0.92
- **F2:** 0.89

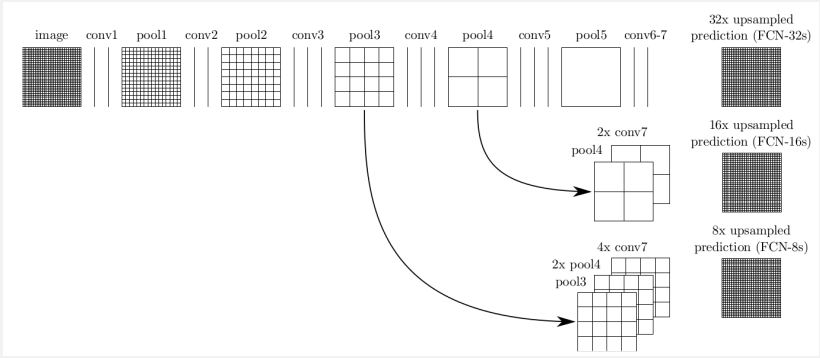
$$F_2 = 5 \cdot \frac{precision \cdot recall}{4 \cdot precision + recall}$$

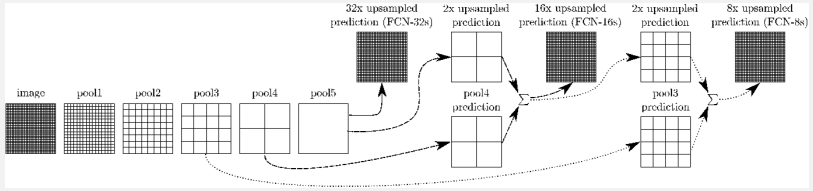
- Plotted the training loss curve as well as the training accuracy curve for 30 epoch
- Biggest decay in loss within first ~ 3-4 iterations
- Accuracy almost saturated after ~ 5-6 iterations

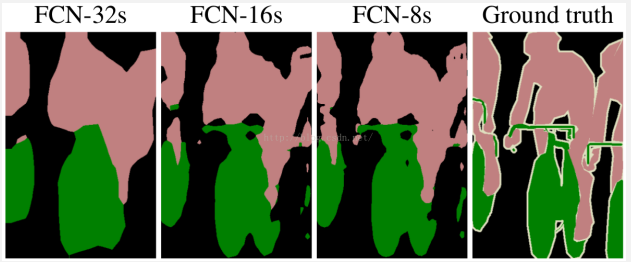
SEMANTIC SEGMENTATION

FULLY CONVOLUTIONAL NETWORKS (FCN)

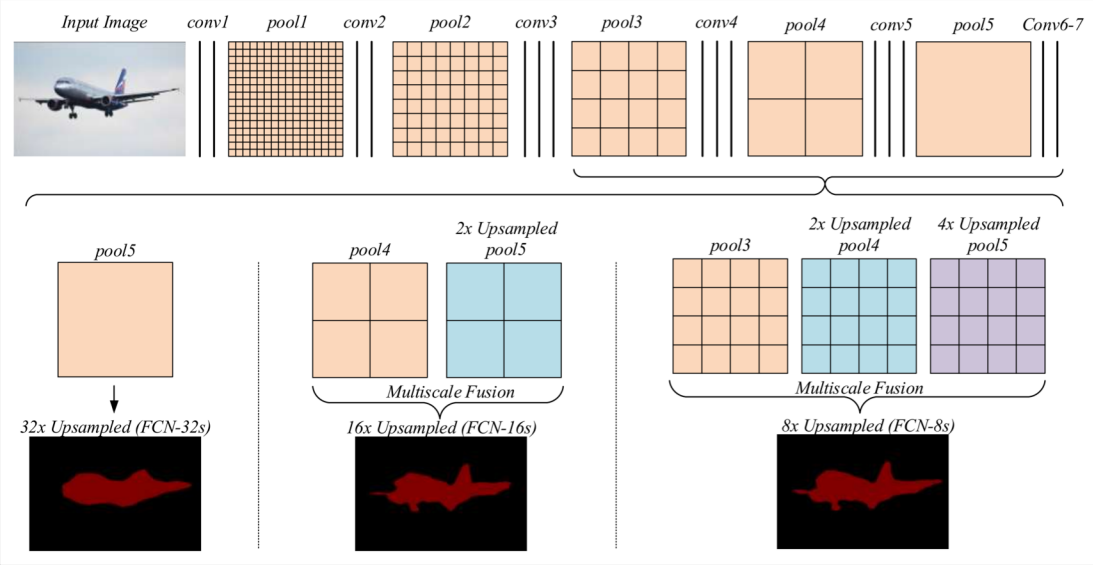
- Built from locally connected layers (i.e. convolution, pooling and upsampling and not dense layers)
- <u>Segmentation networks usually have 2 parts :</u>
  1. **Downsampling path**
     - Capture semantic/contextual information (i.e. used to extract and interpret the context (*what* is present in the image))
  2. **Upsampling path**
     - Recover spatial information (i.e. used to enable precise localization (*where* it is present in the image))
- **Skip connections**
  - a connection that bypasses at least one layer.
  - Used to recover fine-grained spatial information lost in the pooling or downsampling layers
  - used to transfer local information by concatenating or summing feature maps from the downsampling path with feature maps from the upsampling path. Merging features from various resolution levels helps combining context information with spatial information.



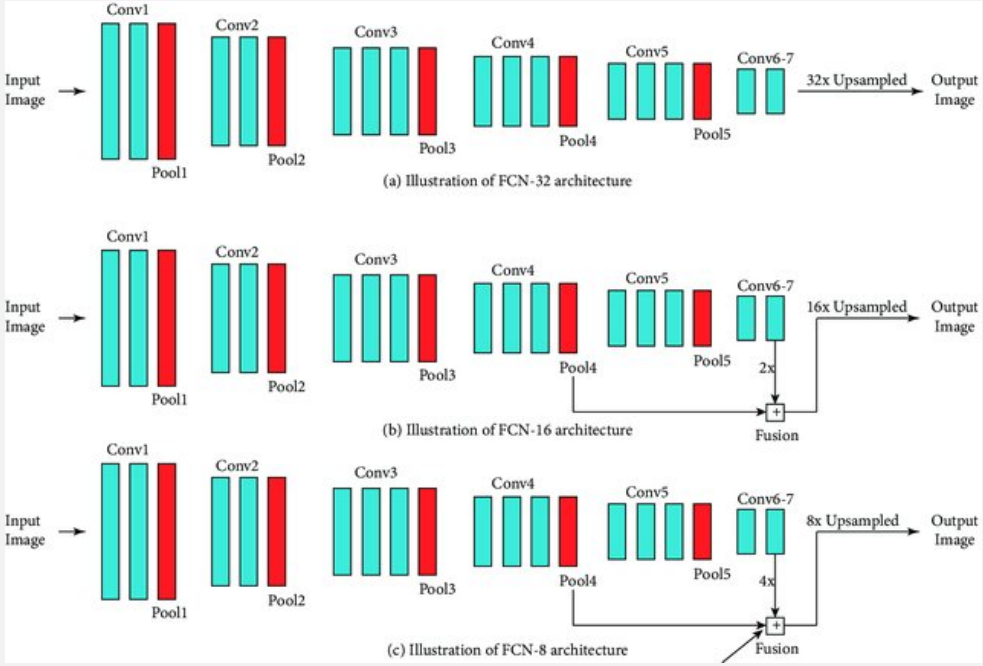




FULLY CONVOLUTIONAL NETWORKS (FCN)



- <u>The 3 FCN architectures:</u>
  - Share the same downsampling path
  - Differ in their respective upsampling paths
  - They use different skip connection layers and strides for the last convolution



1. **FCN-32**
   - Produces the segmentation map directly from *conv7* (using transposed convolution layer with stride 32 I.e.

2. **FCN-16**
   - Sums the 2x upsampled prediction from *conv7* with *pool4* ==> perform 16× upsampling to produce the segmentation map.

3. **FCN-8**
   - Sums the 2x upsampled *conv7* with *pool4* ==> upsamples them, then sums them with *pool3* ==> upsamples final time to produce the segmentation map.

U-NET

- U-Net Model
  - FCN variant (because only contains Convolutional layers and does not contain any Dense layer)
  - Developed by Olaf Ronneberger et al. for Bio Medical Image Segmentation
  - Symmetric shape (which is different from other FCN variants we have seen)
  - Extended to work with fewer training images and to yield more precise segmentations

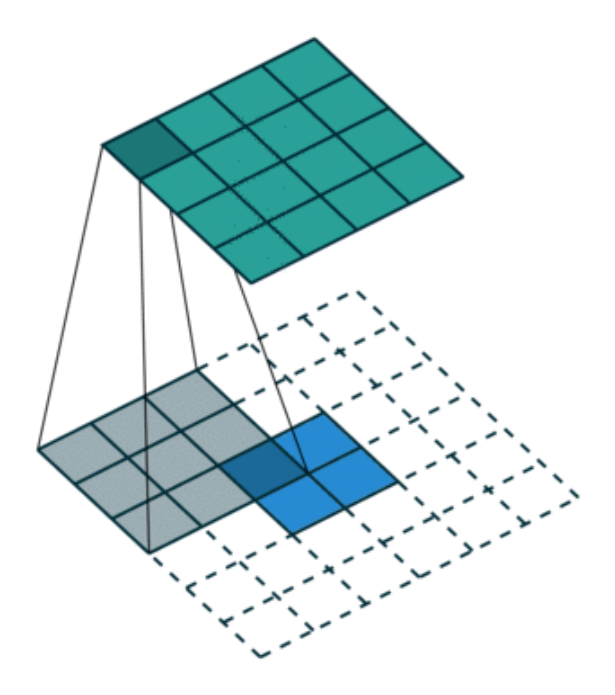
U-NET vs. Standard FCNs

1. FCNs
   - Skip connections apply **summation (i.e. tf.add downsampling feature maps to upsampling layers)**
   - Use **bilinear interpolation** for upsampling so there is *no learnable filter* in the upsampling
2. U-Net
   - U-net is **symmetric**
   - Skip connections apply a **concatenation** operator **instead of a sum (i.e. tf.concat downsampling feature maps to upsampling layers)**
   - Uses **transposed convolutions** (aka *deconvultion* or *fractionally strided convolution*) which are *learnable weight filters* instead of fixed interpolation techniques (I.e. bilinear and other techniques we discussed in the image processing lecture)

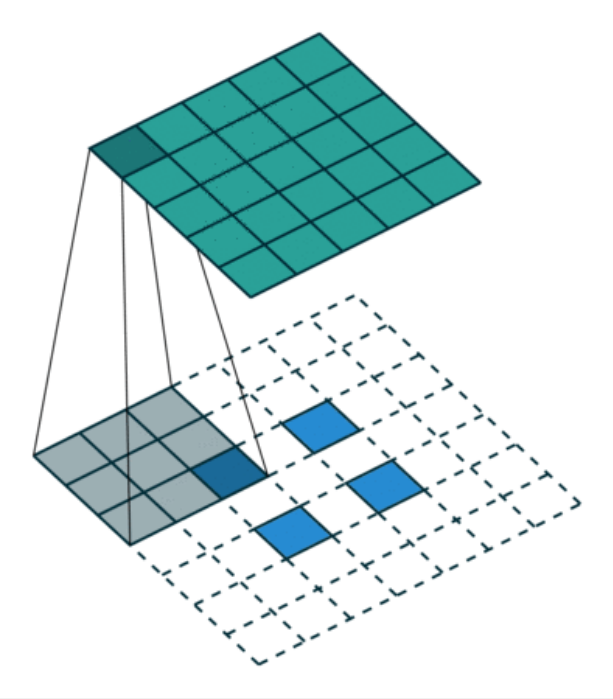
TRANSPOSED CONVOLUTION

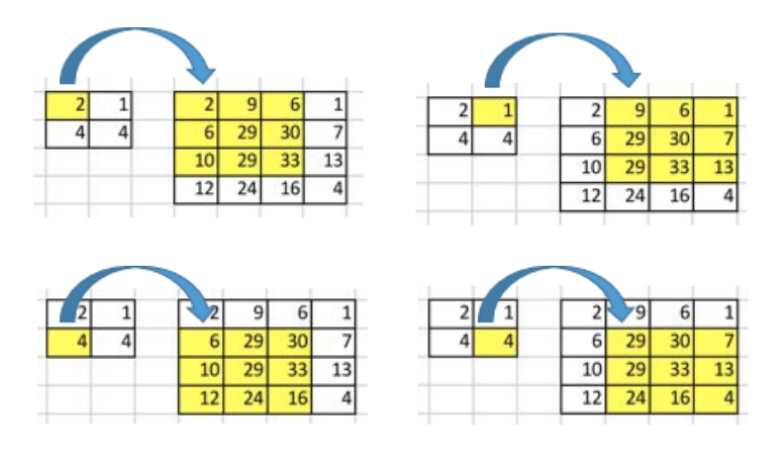
- Transposed convolution is the opposite process of a normal convolution
  - Regular Convolution = many-to-one relationship (i.e. high resolution => low resolution)
  - Transposed Convolution = one-to-many relationship (i.e. low resolution => high resolution)

No padding, no strides, transposed



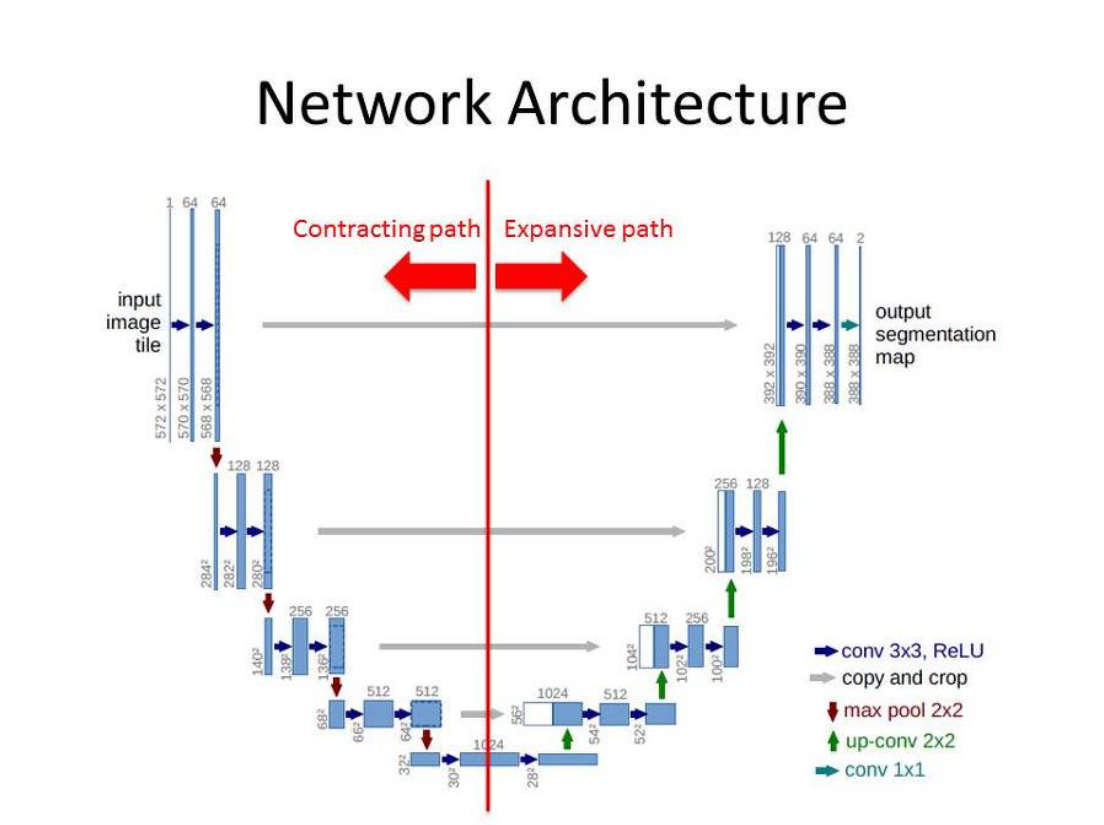
No padding, strides, transposed





Blue maps are inputs, and cyan maps are outputs.

U-NET ARCHITECTURE



Advantages

- General information
  - Combines the location information from downsampling path with contextual information from upsampling path to obtain a general information for segmentation
- No dense layers
  - so images of different sizes can be used as input (since the only parameters to learn on convolution layers are the kernel, and the size of the kernel is independent from input image' size).

1. Downsampling/Contracting

Four block each composed of:

1. 3x3 Convolution Layer + activation function (with batch normalization)
2. 3x3 Convolution Layer + activation function (with batch normalization)
3. 2x2 Max Pooling

2. BottleNeck

Simply 2 convolutional layers (with batch normalization), with dropout.

3. Upsampling/Expanding

Four block each composed of:

1. Deconvolution layer with stride 2
2. Concatenation with the corresponding cropped feature map from the contracting path
3. 3x3 Convolution layer + activation function (with batch normalization)
4. 3x3 Convolution layer + activation function (with batch normalization)

CONCLUSION

CONCLUSION

Image Classification:

- Results from Transfer learning with 'DenseNet169' resulted in an 0.89 F2-score, 0.88 Recall, and 0.91 Precision
- In the future, it would be interesting to implement different networks (e.g., deeper networks) to train the classifier
- Furthermore, it would be interesting to compare the transfer learning model to a CNN that was specifically trained for the ship classification task

Semantic Segmentation:

- Semantic segmentation is used to classify the object class within image at a pixel level.
- Segmentation consists of two phases: fully convolutional layer and upsampling layer. The first phase learns local presentations, and the second phase sums them up together
- We want to use a special type of segmentation called U-net whose downsampling and upsampling have symmetric counterparts