Data Set Augmentation

Sargur N. Srihari
srihari@buffalo.edu

Lecture Notes in Deep Learning: http://www.cedar.buffalo.edu/~srihari/CSE676

Regularization Strategies

1. Parameter Norm Penalties
2. Norm Penalties as Constrained Optimization
3. Regularization and Under-constrained Problems
4. Data Set Augmentation
5. Noise Robustness
6. Semi-supervised learning
7. Multi-task learning

8. Early Stopping
6. Parameter tying and parameter sharing
7. Sparse representations
8. Bagging and other ensemble methods
9. Dropout
10. Adversarial training
11. Tangent methods

Topics in Data Augmentation

1. More data is better
2. Augmentation for classification
3. Caution in data augmentation
4. Injecting noise
5. Benchmarking using augmentation
6. Ex: Heart disease diagnosis using deep learning

More data is better

- Best way to make a ML model to generalize better is to train it on more data
- In practice amount of data is limited
- Get around the problem by creating synthesized data
- For some ML tasks it is straightforward to synthesize data

Augmentation for classification

- Data augmentation is easiest for classification
  - Classifier takes high-dimensional input $\boldsymbol{x}$ and summarizes it with a single category identity y
  - Main task of classifier is to be invariant to a wide variety of transformations
- Generate new samples $(\boldsymbol{x}, y)$ just by transforming inputs
- Approach not easily generalized to other problems
  - For density estimation problem
    - it is not possible generate new data without solving density estimation

Effective for Object Recognition

- Data set augmentation very effective for the classification problem of object recognition
- Images are high-dimensional and include a variety of variations, may easily simulated
- Translating the images a few pixels can greatly improve performance
  - Even when designed to be invariant using convolution and pooling
- Rotating and scaling are also effective

Caution in Data Augmentation

- Not apply transformation that would change the class
- OCR example: ‘b’ vs ‘d’ and ‘6’ vs ‘9’
  - Horizontal flips and 180 degree rotations are not appropriate ways
- Some transformations are not easy to perform
  - Out of plane rotation cannot be implemented as a simple geometric operation on pixels

Injecting noise

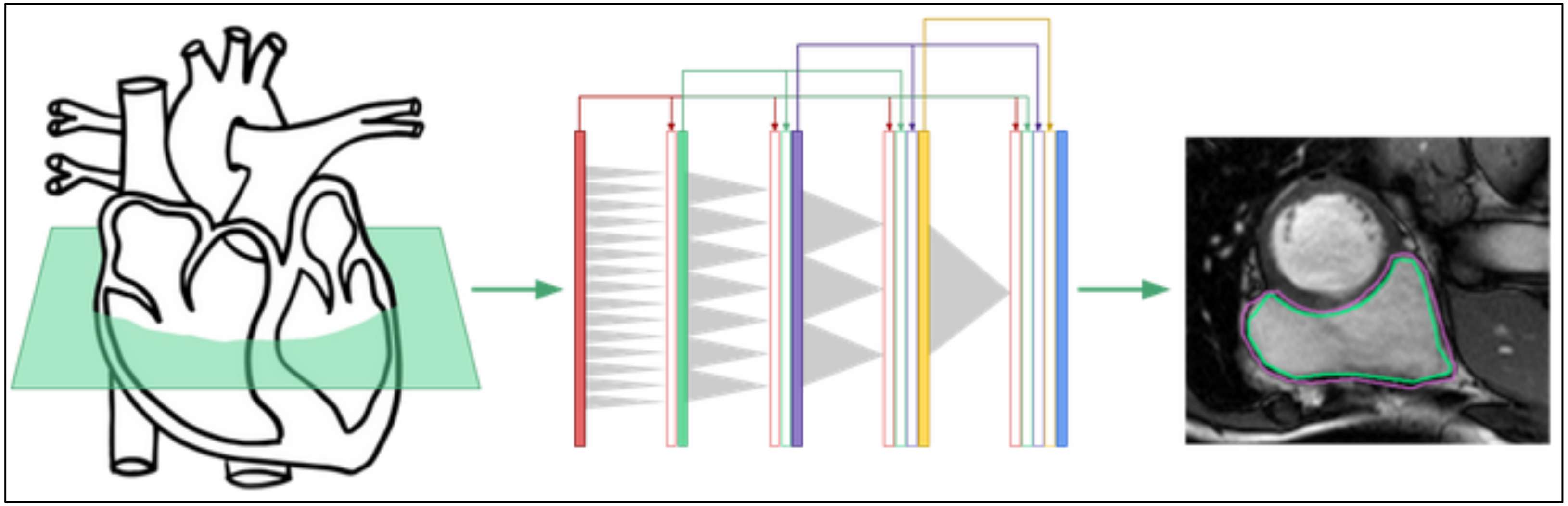
- Injecting noise into the input of a neural network can be seen as data augmentation
- Neural networks are not robust to noise
- To improve robustness, train them with random noise applied to their inputs
  - Part of some unsupervised learning, such as denoising autoencoder
- Noise can also be applied to hidden units
- Dropout, a powerful regularization strategy, can be viewed as constructing new inputs by multiplying by noise

Benchmarking using augmentation

- Hand-designed data set augmentation can dramatically improve performance
- When comparing ML algorithms A and B, same data set augmentation should be used for both
  - If A performs poorly with no dataset augmentation and B performs well with synthetic transformations of the input, reason may be the data set rather than algorithm
- Adding Gaussian noise is considered part of ML while cropping input images is not

Ex: Image segmentation for heart disease

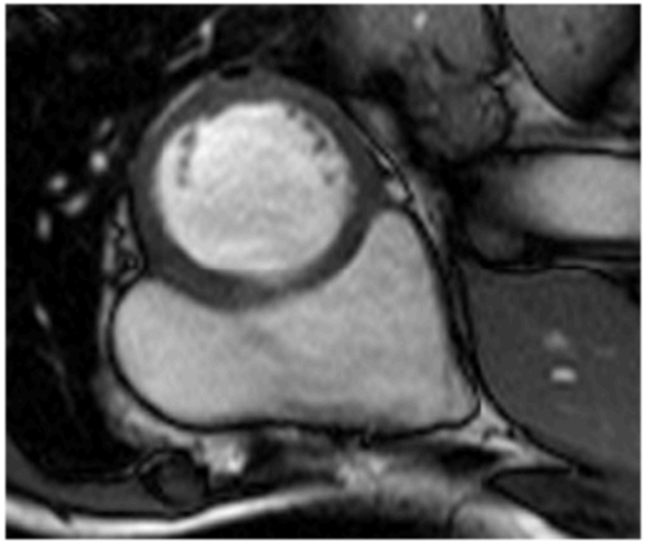
- To determine *ejection fraction*: which measures of how well a heart is functioning
  - After relaxing to its *diastole* so as to fully fill with blood, what percentage is pumped out upon contracting to its *systole?*
    - This metric relies on segmenting right ventricles (RVs) in cardiac magnetic resonance images (MRIs)

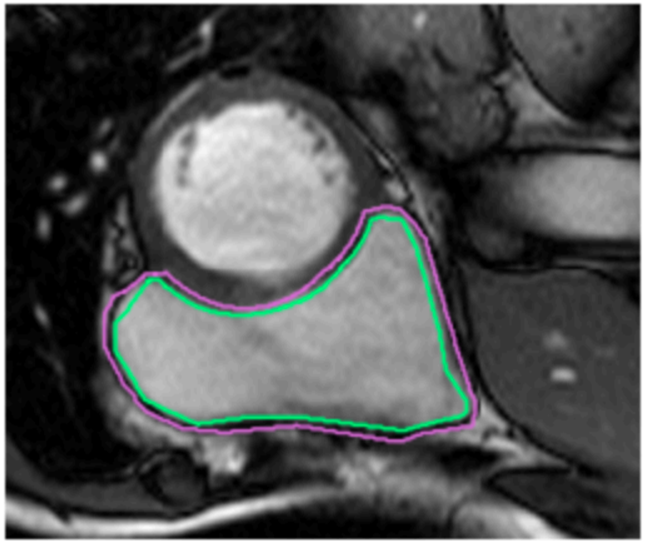


https://blog.insightdatascience.com/heart-disease-diagnosis-with-deep-learning-c2d92c27e730

Problem Description

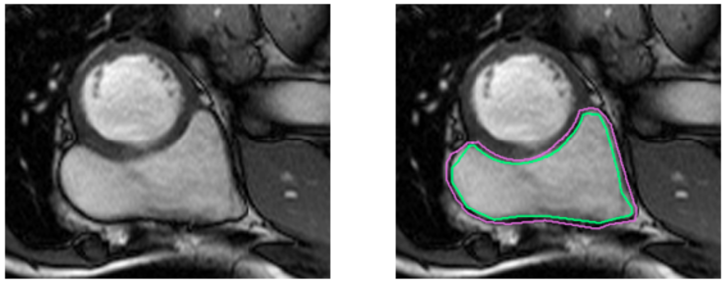
- Develop system to segment RV in cardiac MRI
  - Currently handled by classical image processing
- RV has irregularly shaped thin walls: inner and outer walls (endocardium and epicardium)
  - Manually drawn contours shown:

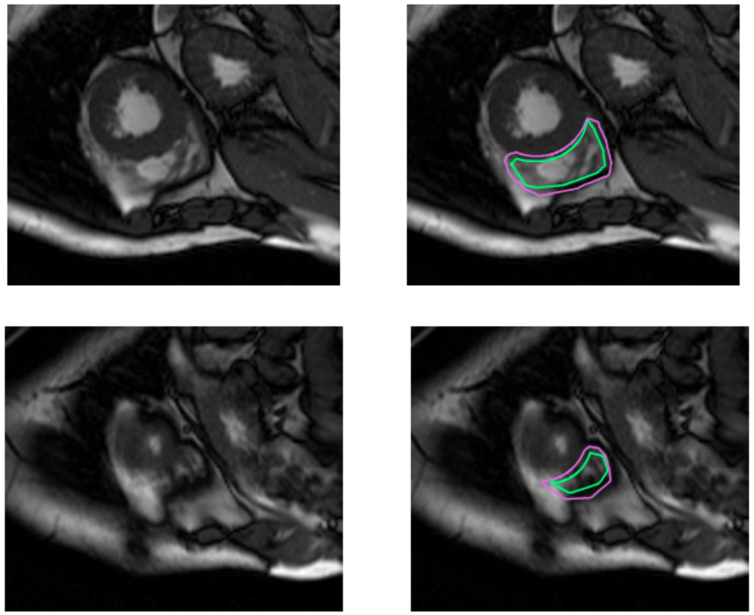




RV segmentation is difficult

- Left ventricle segmentation is easier
  - LV is a thick-walled circle
    - Kaggle 2016 competition
- Right ventricle segmentation is harder
  - Complex crescent shape
  - Easy and hard cases



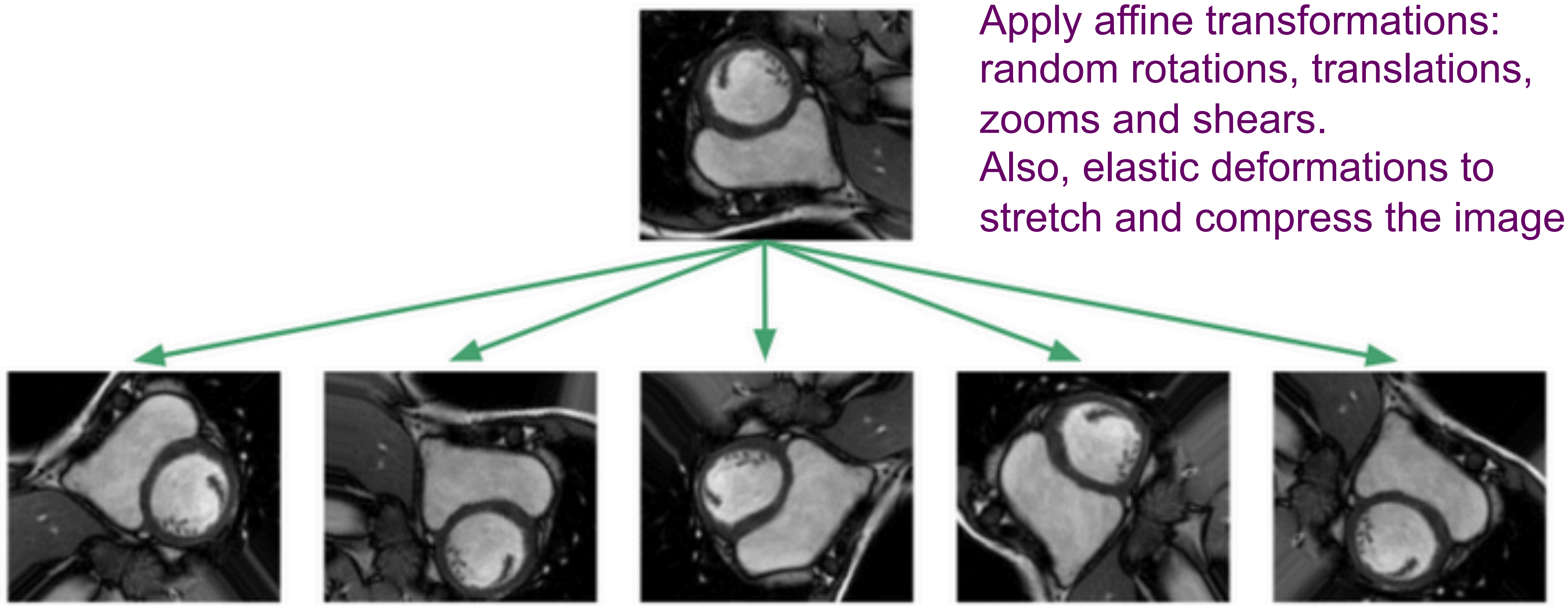


Task: determine whether each pixel is part of RV or not

Need for Data augmentation

- Dataset: 243 physician-segmented images of 16 patients.
  - 3697 additional unlabeled images, useful for unsupervised or semi-supervised techniques
    - Generalization to unseen images would be hopeless!
  - Typical situation in medical settings where labeled data is expensive.

Transformed data



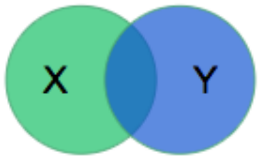
Apply affine transformations: random rotations, translations, zooms and shears.
Also, elastic deformations to stretch and compress the image

Goal: prevent network from memorizing just the training examples, and force it to learn that the RV is a solid, crescent-shaped object in a variety of orientations.

Apply transformations on the fly so the network sees new random transformations during each epoch.

Peformance Evaluation

- Training: 20% of images as validation set
  - RV challenge: separate test set of another 514 MRI images derived from a separate set of 32 patients
- Performance metric
  - The model will output a mask X delineating what it thinks is the RV, and the dice coefficient compares it to the mask Y produced by a physician via:

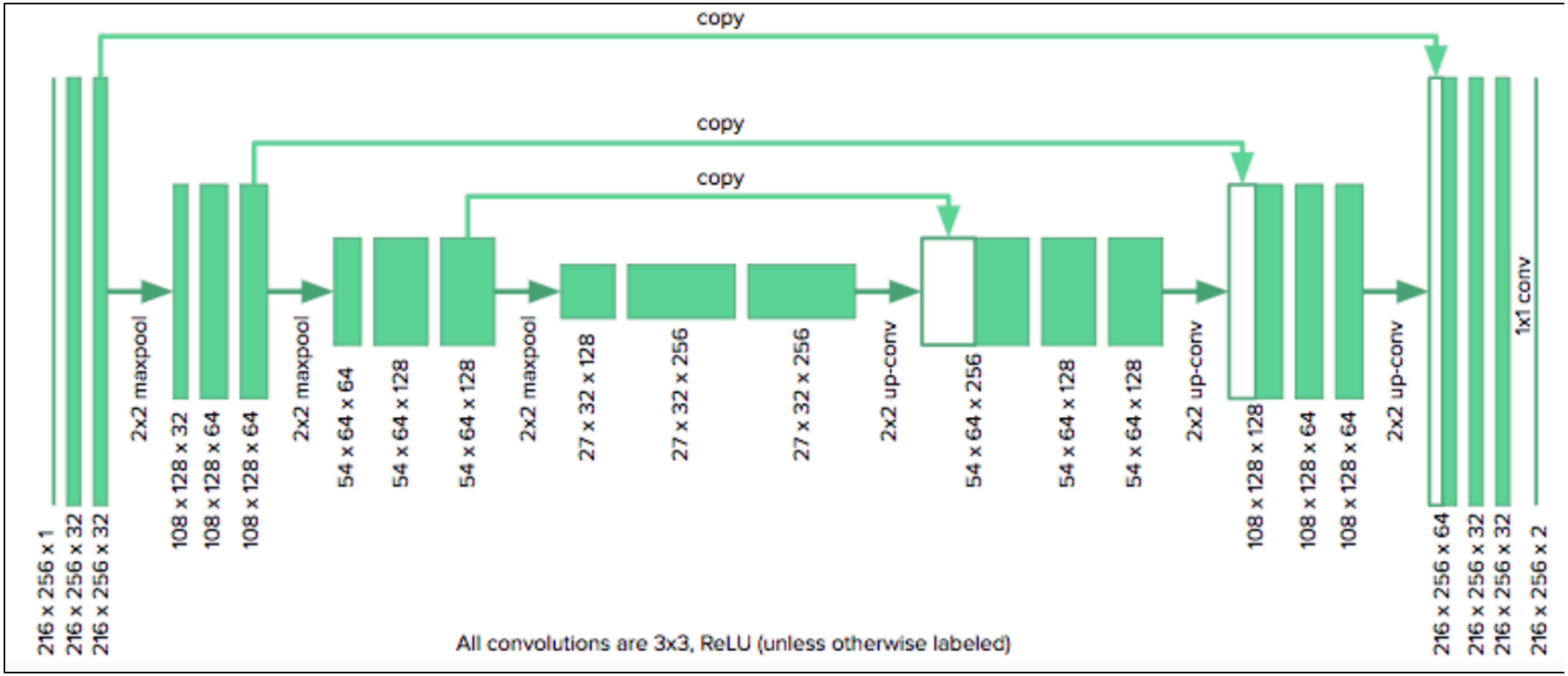


$$\text{dice}(X, Y) = \frac{2X \cap Y}{X + Y}$$

Metric is (twice) the ratio of the intersection over the sum of areas.
It is 0 for disjoint areas, and 1 for perfect agreement.

E.g., model performance is written as $0.82\ (0.23)$, where the parentheses contain the standard deviation.

Deep Learning Architecture



U-net architecture

- Train network with only 30 images using augmentation and pixel-wise reweighting

- It consists of a contracting path, which collapse image into high level features,
- Uses the feature information to construct a pixel-wise segmentation mask.
- Copy and concatenate connections pass information from early feature maps to later portions of the network tasked with constructing the segmentation mask.

Implementation

- Implemented in Keras
  - Code available in Github
    - https://github.com/chuckyee/cardiac-segmentation
- Baseline is fully convolutional network (FCN)
- Endocardium and epicardium performance

Method	Train	Val	Test	Params
Human	–	–	0.90 (0.10)	–
FCN (Tran 2017)	–	–	0.86 (0.20)	~11M
U-net	0.93 (0.07)	0.86 (0.17)	0.77 (0.30)	1.9M
Dilated u-net	0.94 (0.05)	0.90 (0.14)	**0.88 (0.18)**	3.7M
Dilated densenet	0.94 (0.04)	0.89 (0.15)	0.85 (0.20)	**0.19M**

Method	Train	Val	Test	Params
Human	–	–	0.90 (0.10)	–
FCN (Tran 2017)	–	–	**0.84 (0.21)**	~11M
U-net	0.91 (0.06)	0.82 (0.23)	0.79 (0.28)	1.9M
Dilated u-net	0.92 (0.08)	0.85 (0.19)	**0.84 (0.21)**	3.7M
Dilated densenet	0.91 (0.10)	0.87 (0.15)	0.83 (0.22)	**0.19M**

Acknowledgemnts

1. Goodfellow, I., Bengio, Y., and Courville, A., Deep Learning, MIT Press 2016
2. Yee, C-H., "Heart Disease Diagnosis with Deep Learning: State-of-the-art results with 60x fewer parameters" https://blog.insightdatascience.com/heart-disease-diagnosis-with-deep-learning-c2d92c27e730