macadamian

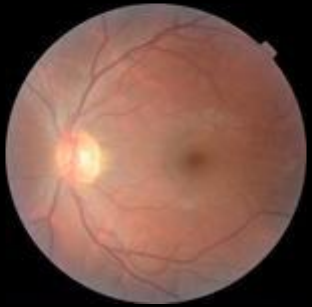
Techsylvania 2018

Image Classification for Medical Diagnostics

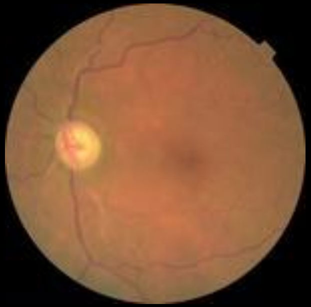
Workshop Goals

- Understand the application of image classification (i.e. “computer vision”) in healthcare
- Understand on a high level what a Convolutional Neural Network (CNN) is and how it works
- Understand how to install and configure your environment to build a CNN using Caffe
- Understand how to build and modify a basic model in Caffe at a high level
- Understand how to train a model with a prepared training data set

Retinopathy



Normal Retina



Retina Showing Signs of Diabetes

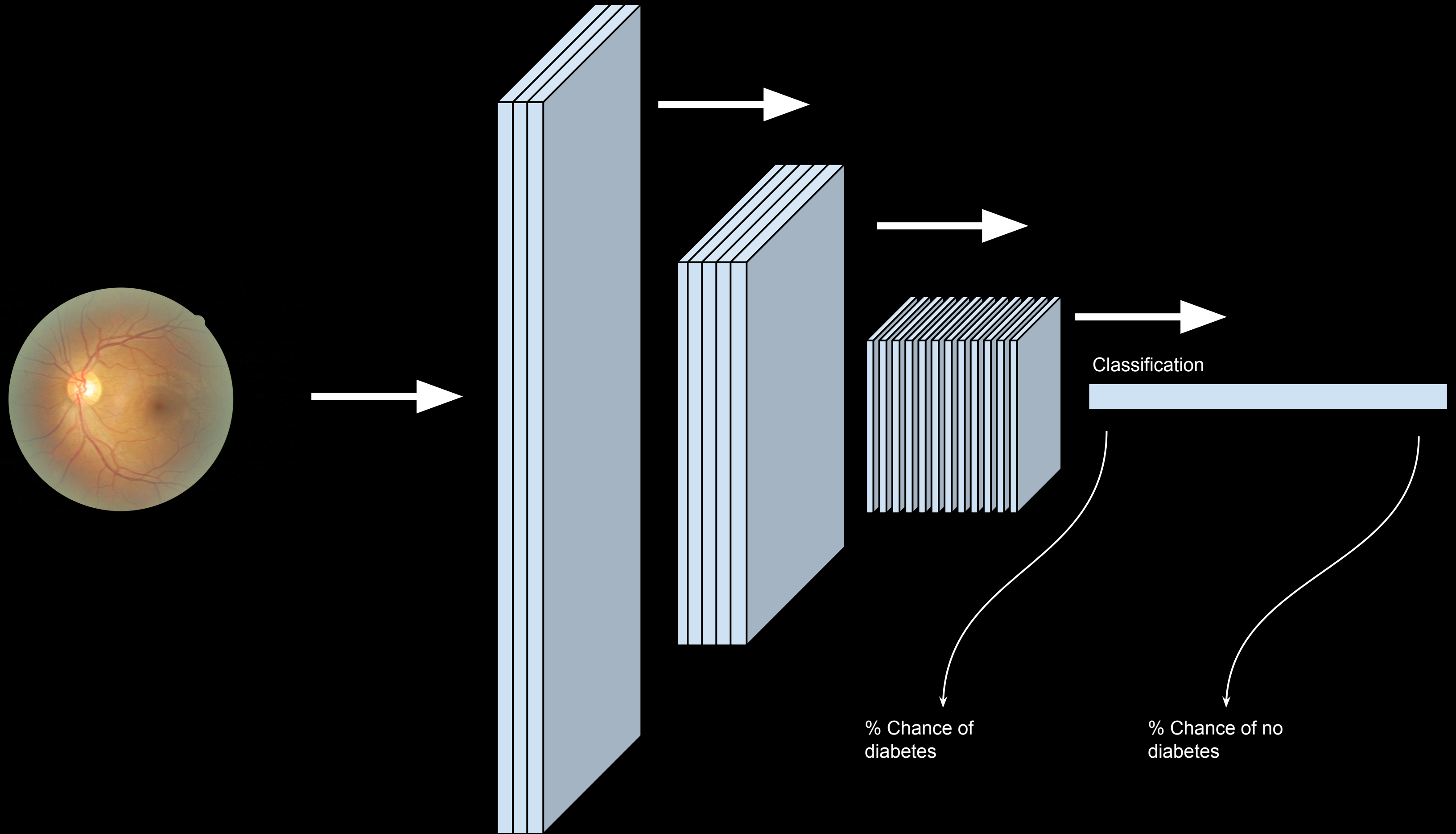
Environment Requirements

- Ubuntu/Debian Linux, Windows, or Mac
- git
- Caffe (GPU version is optional)

Environment Setup

- git clone https://github.com/dcampbell-macadamian/techsylvania2018.git
- Follow steps in “readme.txt” file

Image Classification Using a Convolutional Neural Network (CNN)

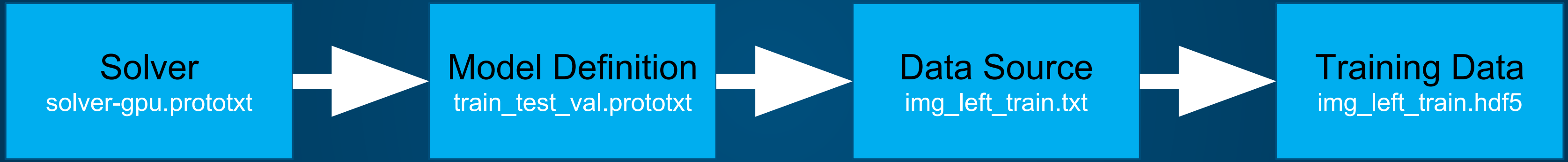


Classification

% Chance of diabetes

% Chance of no diabetes

Caffe Training Configuration



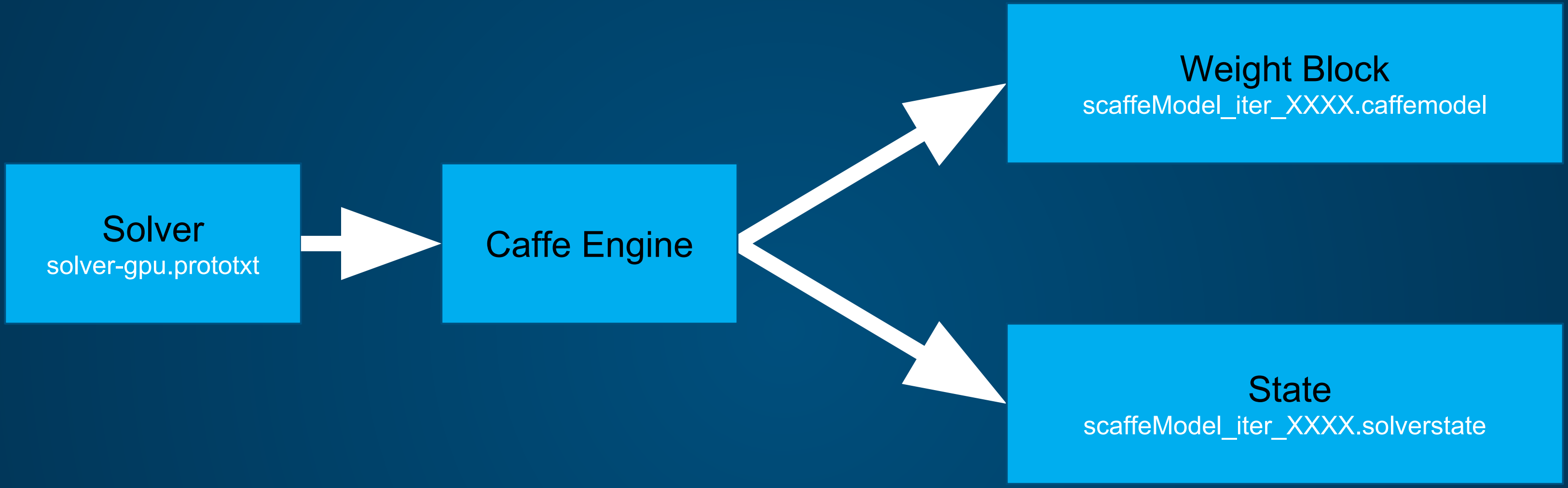
Solver
solver-gpu.prototxt

Model Definition
train_test_val.prototxt

Data Source
img_left_train.txt

Training Data
img_left_train.hdf5

Caffe Training


Solver
solver-gpu.prototxt
Caffe Engine
Weight Block
scaffeModel_iter_XXXX.caffemodel
State
scaffeModel_iter_XXXX.solverstate

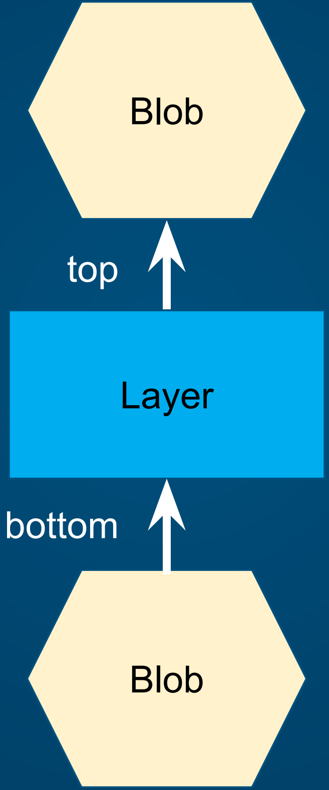
GPU Version

```
caffe train --solver solver-gpu.prototxt -gpu 0
```

No GPU Version

```
caffe train --solver solver-cpu.prototxt
```

Building a Model Caffe Style



Blob

- 4-Dimensional fixed size array of data
- Fixed size of dimensions is known as the shape
    - Example: training data set of images have the shape ***image count X colour channels X height X width***. If the training data contained 100 images that are 150 pixels wide and 150 pixels high and have three colour channels (RGB) then the shape is (100, 3, 150, 150)
- Blobs are created by layers
- Blobs can be read by layers

Layer

- Layers are functions that data (blobs) flow from and/or to
- Several types of layers exist (see http://caffe.berkeleyvision.org/tutorial/layers.html for full catelog of layer types)
- Layers that read blobs read from the bottom
- Layers that write blobs write to the top
- Data flows forward from bottom to top and backwards from top to bottom
- Layers can be optionally turned off for different modes: training or testing

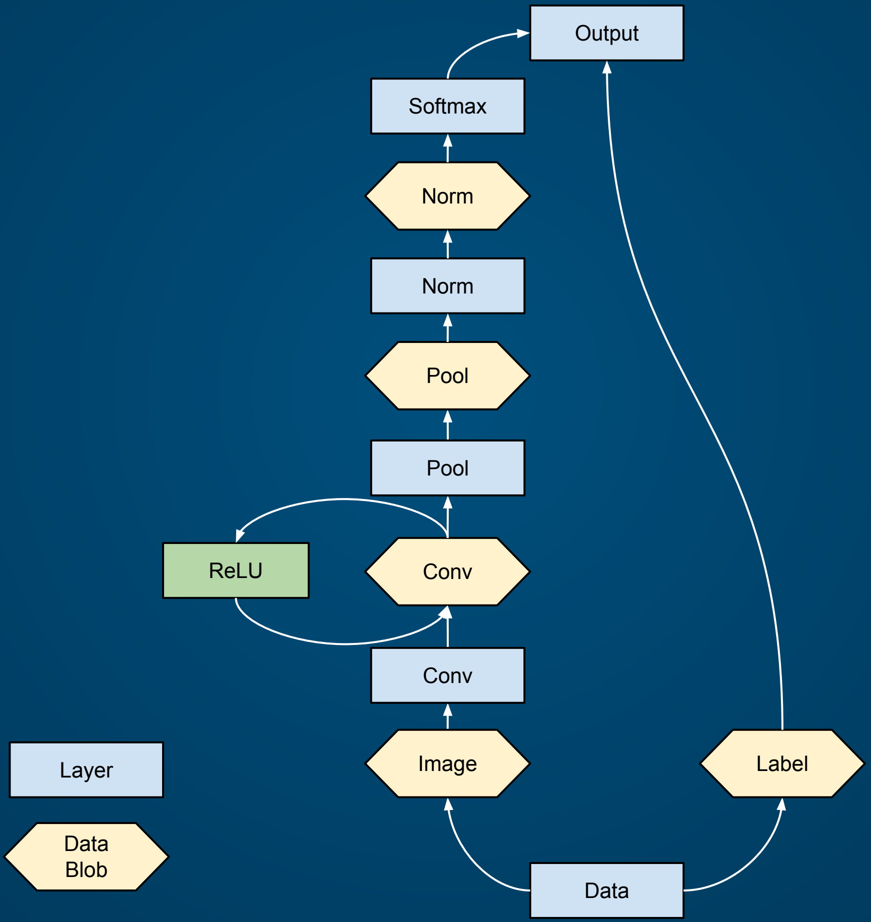
Layer Types Needed for Training and Classification

- HDF5Data
  - Reads data from HDF5 files
  - Creates two blobs: image data and label data
- Convolution
  - Extracts features
  - Reads blob from previous layer
  - Creates a blob (top) with extracted features typically with a different shape than the source (bottom)
- ReLU
  - Stands for Rectified-Linear, which is a math function
  - Needed for training (back propagation)
  - The source (bottom) is always the same shape as the destination (top) blob so top and bottom label can point to the same blob.
- Pooling
  - Used to reduce the shape of the source (bottom) blob
  - Creates a new blob (top) with a reduced shape
  - Prevents overfitting and reduces processing

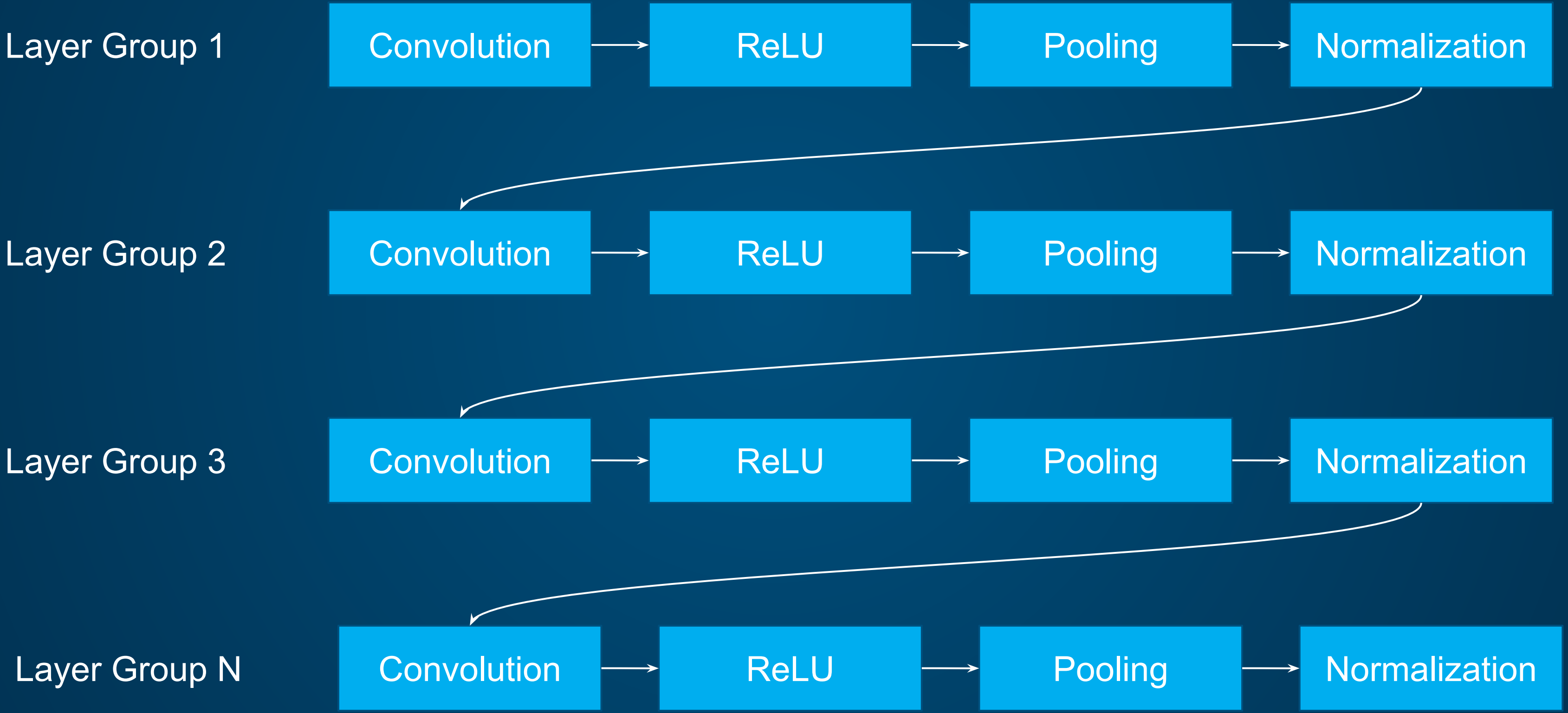
Layer Types Needed for Training and Classification (continued)

- LRN
  - Stands for Local Response Normalization
  - Increases contrast to highlight features
  - Reads (bottom) a blob and creates (top) a normalized blob
- Softmax
  - Takes extracted features from previous layer (bottom) and computes the final classification
  - Classification written (top) to a blob in the same shape as the input (bottom)
- Accuracy
  - Used only for the testing phase (should be turned off for the training phase)
  - Given a test image blob (bottom) and a label blob (bottom), it determines how accurately the network classified a test image
- SoftmaxWithLoss
  - Combines accuracy layer and softmax layer
  - Has two inputs: data blob from previous layer and the label blob
  - Used only for the training phase (should be turned off for the testing phase)

Connecting the Layers



Layer Groupings



Layer Group 1 Convolution → ReLU → Pooling → Normalization

Layer Group 2 Convolution → ReLU → Pooling → Normalization

Layer Group 3 Convolution → ReLU → Pooling → Normalization

Layer Group N Convolution → ReLU → Pooling → Normalization

Model Definition File

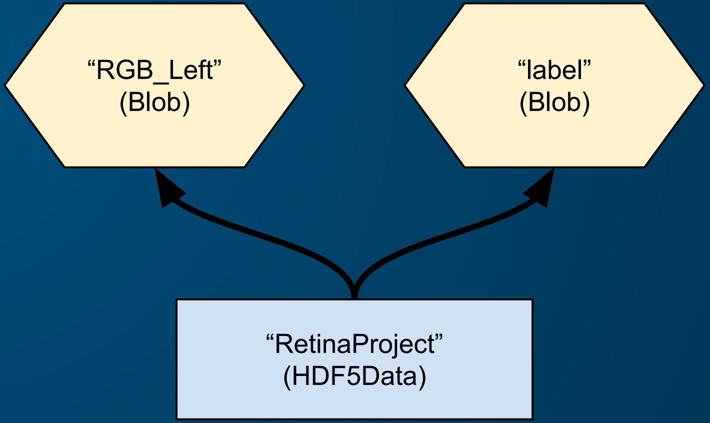
Unique name that identifies the layer

Layer specific parameters

Layer type

```
layer {
  name: "RetinaProject"
  type: "HDF5Data"
  top: "RGB_left"
  top: "label"
  hdf5_data_param {
    source: "testMymodel.txt"
    batch_size: 1
  }
}
```

Name of output blob



Exercise #1: Train a CNN using existing model

Exercise #2: Modify existing model and run training

Questions

David Campbell
Email: dcampbell@macadamian.com
Linkedin: https://www.linkedin.com/in/david-campbell-05b0b813

Thank you.