White Blood Cell Identification System Based on Convolutional Neural Networks

Richard Anderson, Yuanwei Jin+, Enyue Lu**

**Department of Mathematics & Computer Science, Salisbury University, Salisbury, MD;*

+Department of Engineering and Aviation Sciences, University of Maryland Eastern Shore, Princess Anne, MD

Abstract

Background and Objectives: White blood cells (WBCs) differential counting yields valued information about human health and disease. The current developed automated cell morphology equipment performs differential counts which are based on blood smear image analysis. Previous identification systems for WBCs consist of successive dependent stages; pre-processing, segmentation, feature extraction, feature selection, and classification. There is a real need to employ deep learning methodologies so that the performance of previous WBCs identification systems can be increased. Classifying small limited datasets through deep learning systems is a major challenge and should be investigated.

Methods: We propose a new identification system for WBCs based on deep convolutional neural networks. Methodologies based on transfer learning are followed: transfer learning based on fine-tuning of existing deep networks. Specifically, we have used the VGG16 architecture that has been trained on the ImageNet dataset. Moreover, we compare this performance to that of other transfer learning methods. The network is trained on an augmented dataset of 12,500 WBC images and tested on a set of 2503 unique images.

Results: During our experiments, different public WBCs datasets have been used which contain 5 healthy WBCs types. The overall system accuracy achieved by the proposed method is ~99%, which is more than different transfer learning approaches, as well as the previous traditional identification system.

Conclusion: a new WBCs identification system based on deep learning theory is proposed and a high performance convNet can be employed as a pre-trained network..

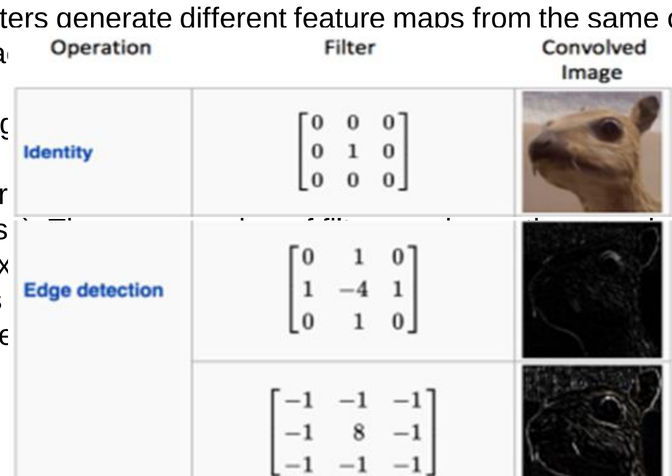
Convolutional Neural Networks

Convolutional Neural Networks (**ConvNets** or **CNNs**) are a category of Neural Networks that have proven very effective in areas such as image recognition and classification. ConvNets have been successful in identifying faces, traffic signs, and thousands of objects apart from powering vision in robots and self driving cars. There are four main operations in ConvNets:

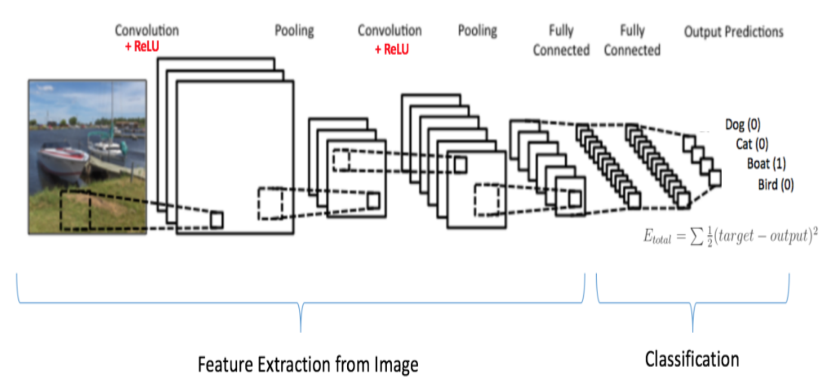
1. Convolution0
2. Non Linearity (ReLU)
3. Pooling or Sub Sampling
4. Classification (Fully Connected Layer)

ConvNets derive their name from the "convolution" operator. The primary purpose of Convolution in case of a ConvNet is to extract features from the input image. Convolution preserves the spatial relationship between pixels by learning image features using small squares of input data. We will not go into the mathematical details of Convolution here, but will try to understand how it works over images.

A filter slides over the input image (convolution operation) to produce a feature map. The convolution of another filter, over the same image gives a different feature map. It is important to note that the Convolution operat- -ion captures the local dependencies in the original image. These two differ- -ent filters generate different feature maps from the same original image. In pra ts own during the training eters such as number of filter he training proces age features get ex t recognizing patterns in unse



Convolutional Neural Networks cont'd



The overall training process of the Convolution Network may be summarized as below:

Step 1: We initialize all filters and parameters / weights with random values

Step 2: The network takes a training image as input, goes through the forward propagation step (convolution, ReLU and pooling operations along with forward propagation in the Fully Connected layer) and finds the output probabilities for each class. Since weights are randomly assigned for the first training example, output probabilities are also random.

Step 3: Calculate the total error at the output layer (summation over all 4 classes)

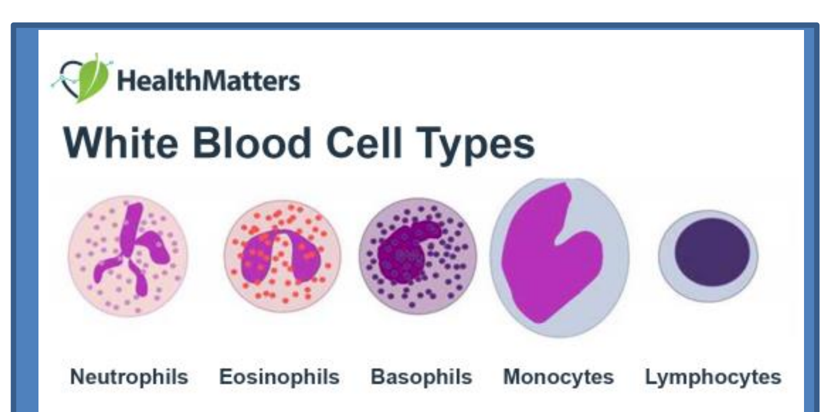
Total Error = ∑ ½ (target probability – output probability) ²

Step 4: Use Backpropagation to calculate the gradients of the error with respect to all weights in the network and use gradient descent to update all filter values / weights and parameter values to minimize the output error. The weights are adjusted in proportion to their contribution to the total error. This means the network has learnt to classify this particular image correctly by adjusting its weights / filters such that the output error is reduced. Parameters like number of filters, filter sizes, architecture of the network etc. have all been fixed before Step 1 and do not change during training process – only the values of the filter matrix and connection weights get updated.

Step5: Repeat steps 2-4 with all images in the training set.

Background Information on White Blood Cells

White blood cells are an important part of your body's immune system. They're responsible for protecting your body against infections and invading organisms. You have five types of white blood cells:



Each of these can be affected in different ways if you have a particular condition or disease. Your doctor may request a WBC count and differential if they suspect you have one of several conditions, including:

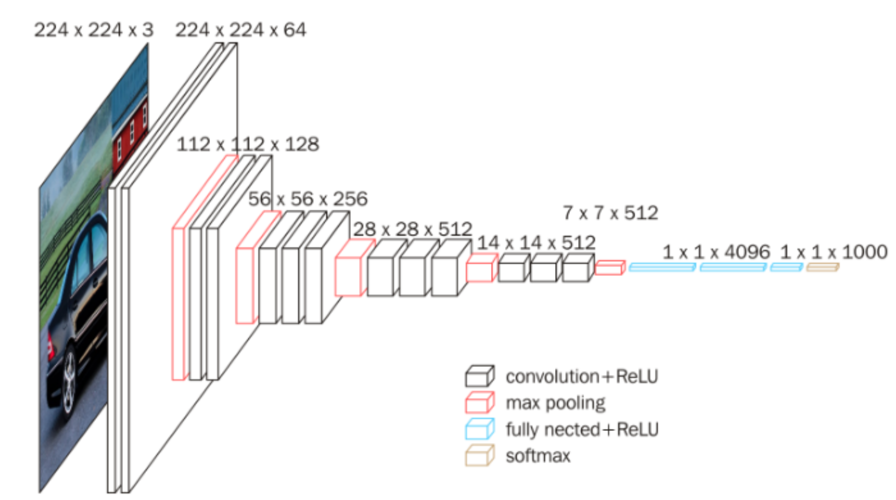
-anemia

-Infection

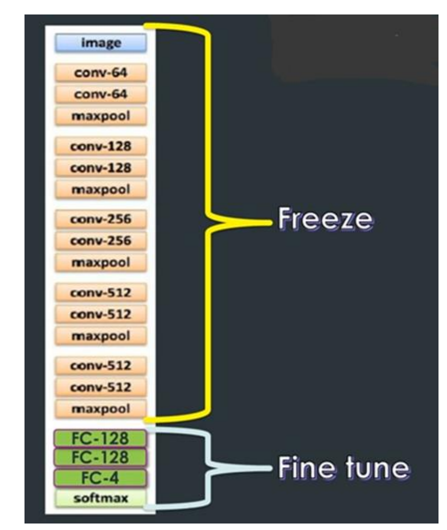
-leukemia

Proposed Method

So far, I have been working on using transfer learning with the VGG16 architecture. The VGG16 is a convolutional neural network architecture named after the Visual Geometry Group from Oxford, who developed it. It was used to win the ILSVR (ImageNet) competition in 2014.

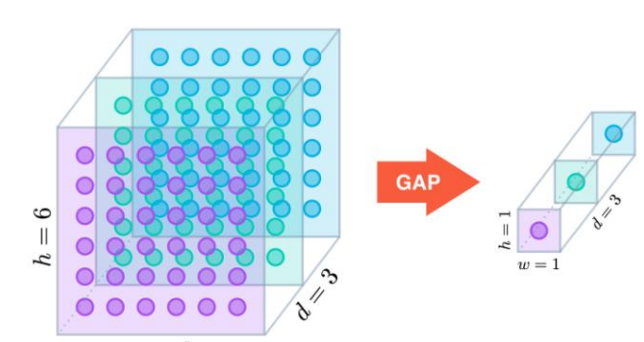


For this project, I implemented the following modified version of the VGG16:



Global Average Pooling

Global average pooling (GAP) is used to minimize overfitting by reducing the total number of parameters in the model. Similar to max pooling layers, GAP layers are used to reduce the spatial dimensions of a three-dimensional tensor. However, GAP layers perform a more extreme type of dimensionality reduction, where a tensor with dimensions h×w×d is reduced in size to have dimensions 1×1×d. GAP layers reduce each h×w feature map to a single number by simply taking the average of all hw values.

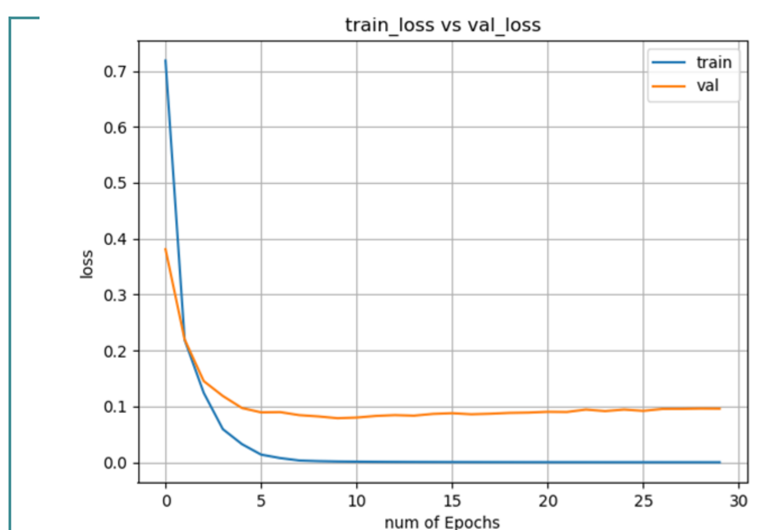


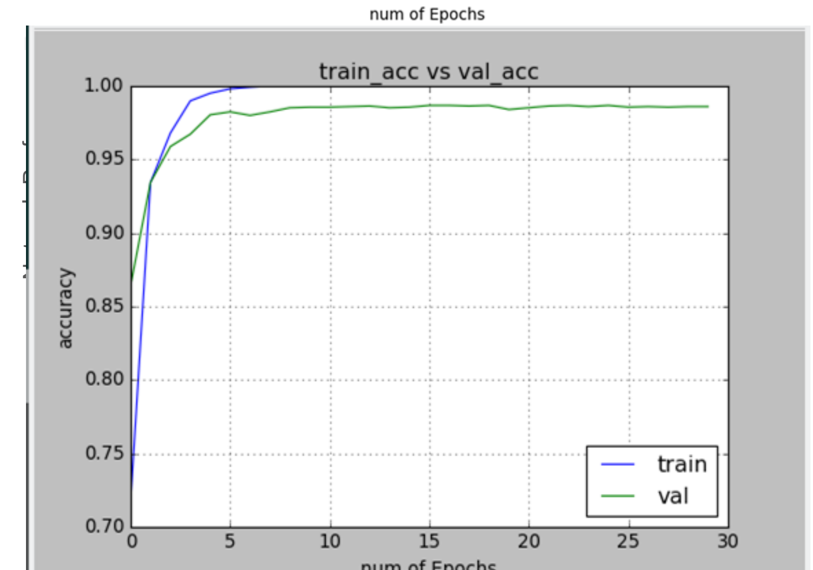
This approach resulted in a greatly reduced number of parameters in the hidden layers. There were around 60,000 parameters in the FC layers of the final version as opposed to over 120 million in the original version! This resulted in a much "simpler" function for our model to learn. As a result, we were able to greatly reduce overfitting. Before GAP, the model achieved around 88% accuracy on the test set. Afterwards, the model improved to approximately ~99% accuracy.

```
flatten (Flatten)            (None, 25088)          0
fc1 (Dense)                  (None, 4096)           102764544
fc2 (Dense)                  (None, 4096)           16781312
predictions (Dense)          (None, 1000)           4097000
=====================================================
Total params:
Trainable par    : 138,357,544
Non-trainable
```

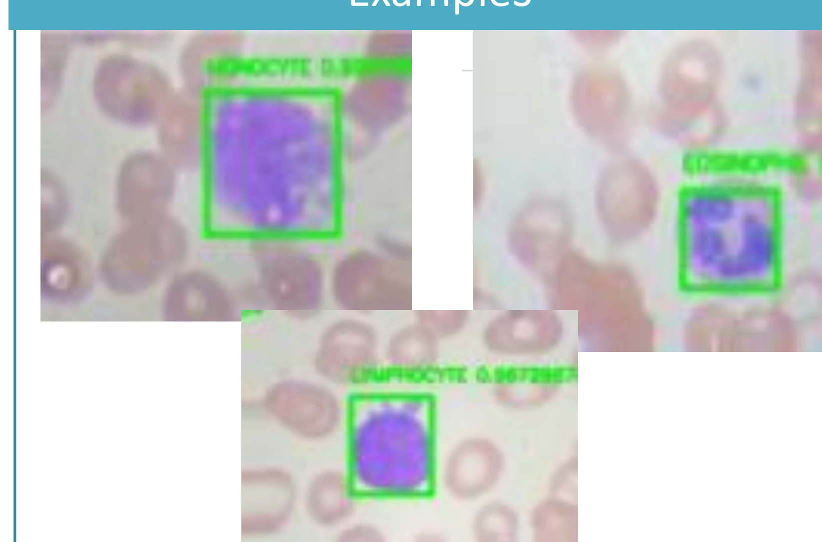
```
global_average_pooling2d_1 ( (None, 512)            0
fc1 (Dense)                  (None, 100)            51300
fc2 (Dense)                  (None, 100)            10100
output (Dense)               (None, 4)              404
=====================================================
Total params: 14
Trainable param 14,776,492
Non-trainable pa
```

Results





Examples



Future Work

Future work for this project includes gathering more data in order to have a more robust network and a higher degree of accuracy. On top of this, we might be able to find "lighter" architectures that may contain even less layers/parameters. Having a faster algorithm could potentially be beneficial in the case that the user does not have a GPU and there are many slide images to be processed. Additionally, this algorithm will become the backbone of another algorithm that specializes in cell counting. Finally, we would like to create visualizations in order to better understand the network's performance.

References

Shahin, A. I., Guo, Y., Amin, K. M., & Sharawi, A. A. (November 01, 2017). White Blood Cells Identification System Based on Convolutional Deep Neural Learning Networks. *Computer Methods and Programs in Biomedicine,* 5.)

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *CoRR*. Retrieved June 20, 2018, from https://arxiv.org/abs/1609.04747

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556.