Sparse Representation Classifier for Artificial Intelligent Microscopy of Blood Cell Subtypes Using a Tensor Processing Unit

Ashwin Parthasarathy

Summer Research (HH)

Dr. Iris Thompson

Period 10

Abstract

Microscopic blood cell analysis is a critical activity in the pathological analysis of blood-based diseases, meaning automated methods to classify cell subtypes have important medical applications. A sparse representation classifier (SRC) is a type of algorithm that utilizes patterns to recognize characteristics of data in a low-dimensional space, which is particularly useful for analyzing microscopy, and typically requires less data to train the model. By running this algorithm on a tensor processing unit, these machine learning tasks can be further optimized using this specialized hardware. The purpose of this experiment was to develop a sparse representation model running on a tensor processing unit that could identify and characterize patient blood samples for blood-based diseases during real-time microscopy using fewer data examples for training compared to a convolutional neural network (CNN). Augmented images of blood cells from four different cell subtypes were downloaded and two separate models, a SRC and a standard CNN, were built to compare the number of data examples needed for accurate diagnosis**.** The results demonstrate that when the SRC was run for 500 training images per cell subtype, the accuracy of the model was 85.1%, which is greater than the accuracy of 84.9% for the CNN with 1500 training images per cell subtype, approximately three times the number of training images. This demonstrates that by using a SRC and a tensor processing unit, the amount of blood data and thus the cost and time associated with blood-based disease diagnosis using cell subtypes can be minimized.

Introduction

Purpose

The purpose of this experiment is to develop a sparse representation model running on a tensor processing unit that can identify and characterize patient blood samples for blood-based diseases during real-time microscopy using fewer data examples for training compared to a convolutional neural network. Automated methods to detect and classify blood cell subtypes have important medical applications for blood-disease diagnosis, and by comparing machine learning models to determine which network requires the least number of blood samples for training, it is possible to minimize the amount of blood data and thus the cost associated with blood disease diagnosis.

Background Research

White blood cells, also known as leukocytes, are components of blood produced in bone marrow and stored in lymphatic and blood cell tissues. Blood is made up of red and white blood cells, platelets, and plasma, and although white blood cells account for only 1% of this fluid, they play a pivotal role in protecting the human body from illness and disease. These cells fight viruses and bacteria within the bloodstream, destroying harmful substances to prevent illness. White blood cells develop stem cells that can mature into five major types belonging to two main classes: Granulocytes, which include Neutrophils, Eosinophils, and Basophils, and Agranulocytes, which include Lymphocytes and Monocytes. The main difference between granulocytes and agranulocytes arises from the presence of specific granules in the cell's cytoplasm, which also results in varying functions. For example, neutrophils are phagocytic cells that kill microbes, eosinophils are involved in inflammatory processes like parasitic infections and allergic diseases, and basophils increase inflammation. On the other hand, lymphocytes recirculate through tissue via lymphatic vessels and monocytes differentiate into the various macrophages of the mononuclear system (Sadafi et al., 2019).

Identifying cell subtypes is important for hematologists and doctors to avoid medical risks and specify the right therapies for blood-based diseases. Using optimized ways for diagnosis will facilitate and speed up the diagnosis of blood-based diseases using the blood cells images, such as blood smears and other microscopic data. Analysis of blood smears and microscopic data is not the only one technique for cell subtype analysis, but it is the most common and most powerful way. Microscopic white blood cell analysis is a critical activity in the pathological analysis of blood-based diseases, meaning automated methods to detect and classify white blood cell subtypes have important medical applications for disease diagnosis (Ghane et al., 2019).

Deep learning is a sub-field of machine learning that utilizes artificial neural networks, structures that enable computers to learn from observational data using layers of neuron-like nodes, a process similar to how human brains analyze information. Deep learning uncovers intricate structure in large datasets by using the backpropagation algorithm, specifying how the machine should update the internal parameters it uses to compute the representation in each layer from the representation in the previous layer. Deep learning architecture has dramatically improved the state-of-the-art in speech recognition, visual object recognition, and image recognition, which is directly applicable to medical imaging (LeCun et al., 2015).

A sparse representation intends to represent a signal with as few as possible significant coefficients, compressing instances of data during these examples. However, it is frequently noticed that a great compression rate can be obtained with almost unnoticeable loss of information, particularly for information obtained within a low-dimensional space. By using sparse representation, we can concisely represent the data and easily extract the valuable information from the data. Sparse representation classification is a powerful technique for pixelwise categorization of images and is increasingly being used for a wide variety of image analysis tasks. A sparse representation classifier is a type of algorithm that utilizes patterns to recognize characteristics of data in a low-dimensional space and typically requires less data to train the model, which is especially useful for analyzing real-time microscopy (Plenge et al., 2015).

Deep learning models typically require large amounts of data, which is difficult to obtain for expensive medical imaging procedures. However, by utilizing a sparse representation architecture, it is possible to address the problems of labeled data scarcity through the compression and extraction of information achieved by this classifier. This method of data processing is different from that of a standard convolutional neural network for deep learning, so by comparing the effectiveness of these models at the task of disease diagnosis, it is possible to see which algorithm can be used to minimize the amount of training data necessary. Furthermore, a tensor processing unit, which is an AI accelerator integrated circuit developed by Google specifically for neural network machine learning, is a piece of hardware that can be particularly useful for comparing the effectiveness of two models over many iterations. By running these algorithms on a tensor processing unit, these machine learning tasks can be further optimized using this specialized equipment. (Zhu et al., 2016).

Automated methods to detect and classify blood cell subtypes have important medical applications for blood-disease diagnosis, and by comparing machine learning models to determine which network requires the least number of blood samples for training, it is possible to minimize the amount of blood data and thus the cost associated with blood disease diagnosis.

Hypothesis

If a sparse representation classifier is developed running on a tensor processing unit, then the model will be able to identify and characterize white blood cell subtypes for blood-based diseases during real-time microscopy using half the number of data examples for training compared to a convolutional neural network.

Methods

Materials

1. Computer
2. Internet Connection
3. Kaggle Image Dataset of Blood Cell Subtypes (Eosinophil, Lymphocyte, Monocyte, and Neutrophil)
4. Google Colab
5. Tensor Processing Unit
6. Jupyter Notebook IDE
7. Computer Terminal
8. Tensorflow Python Library

Procedure

Data Pre-Processing:

1. Download the 12,500 augmented images of blood cells with accompanying cell subtype labels from Kaggle with a JPEG format
2. Organize the total images into four different folders for each of the four different cell subtypes (Eosinophil, Lymphocyte, Monocyte, and Neutrophil), with approximately 3,125 images belonging to each cell type.
3. Standardize all of the images so they follow the specified size of 256 by 256 pixels.
4. From the folder of images for each of the four different cell types, randomly select 85% for training the sparse representation model and the convolutional neural network, while the other 15% will be used for validation of these two networks.

Building and Training the SRC Model and Convolutional Neural Network:

5. Build the sparse representation model as a three-layer dense network that implements a sequential architecture. Each dense layer of the sparse representation classifier should apply batch normalization and implement LeakyReLU.
6. Build the convolutional neural network model with four layers, three of which are dense layers. The neural network first contains a flattened layer and then three dense layers implementing LeakyReLU.
7. Implement a steepest gradient descent approach based on the Adam optimization algorithm with step length 1 x 10^-3 to optimize the parameters of both the sparse representation model and the convolutional neural network.

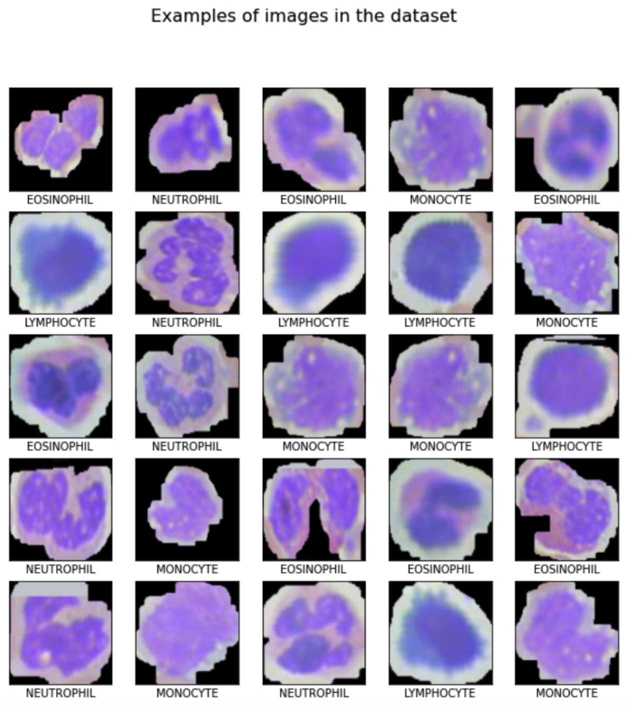
8. Train both the sparse representation model and the convolutional neural network using 500 randomly selected training images from each of the four cell subtypes, a total of approximately 2,000 training images.

Running the Validation Set Through the Models:

9. Run the remaining 15% of validation images through the trained sparse representation model and the trained convolutional neural network, recording the accuracy of each model.
10. Repeat steps 8 and 9, increasing the number of randomly selected images used for training by 1,000 each time until all of the training images have been used.
11. Record and compare the accuracy of each model when trained for 500, 1,500, and 2,500 training images per cell subtype to determine which model minimizes the amount of data needed to train a model capable of accurately classifying the four different blood cell subtypes.

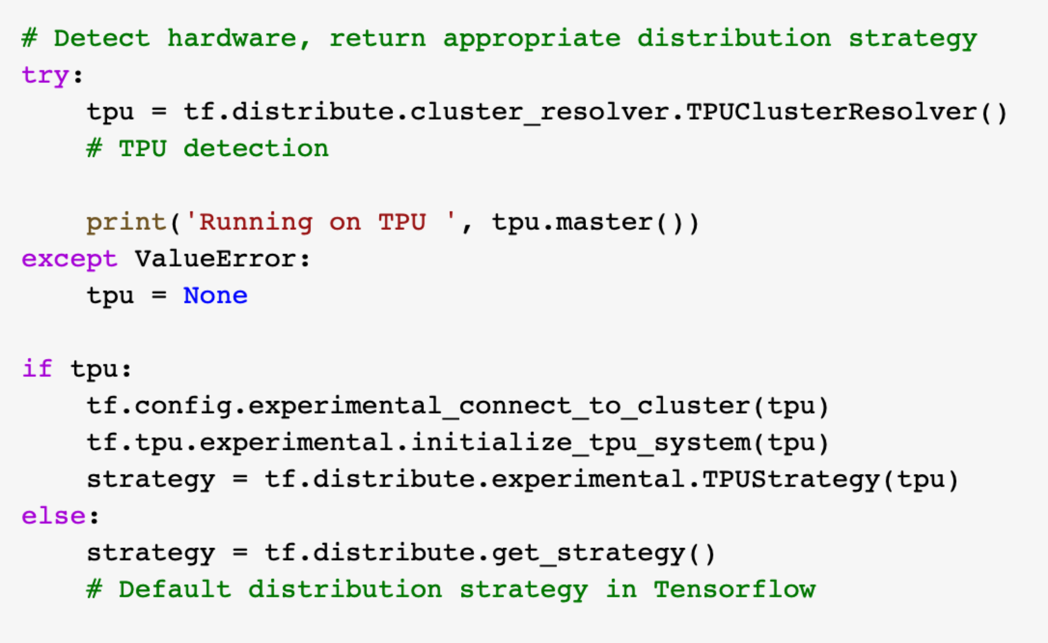
Results

Pictures



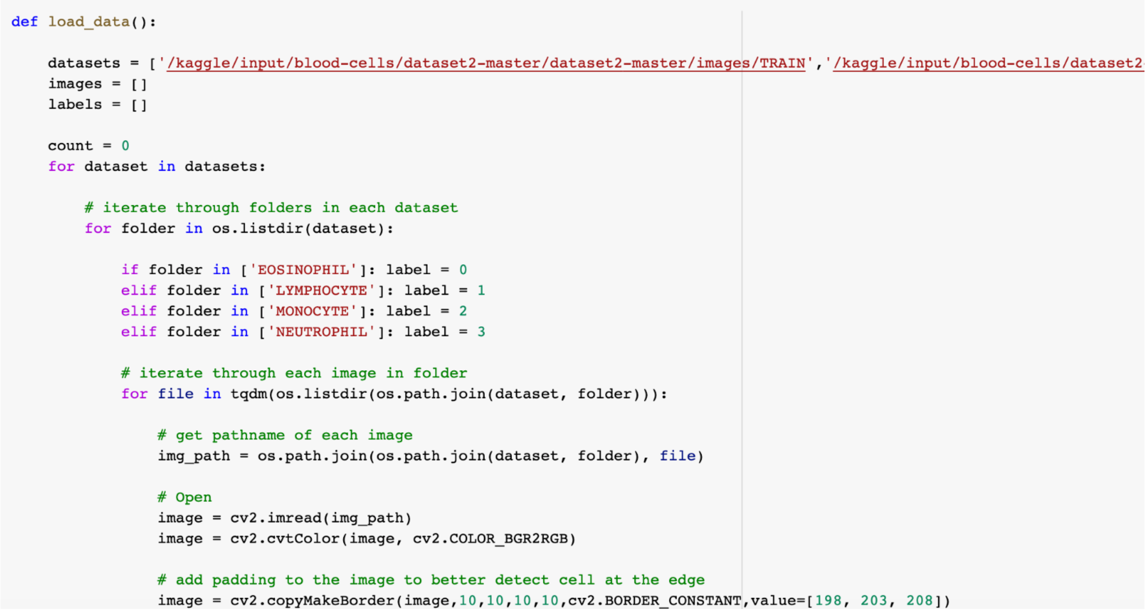
Examples of images in the dataset

White blood cell image dataset with labeled examples of each cell subtype



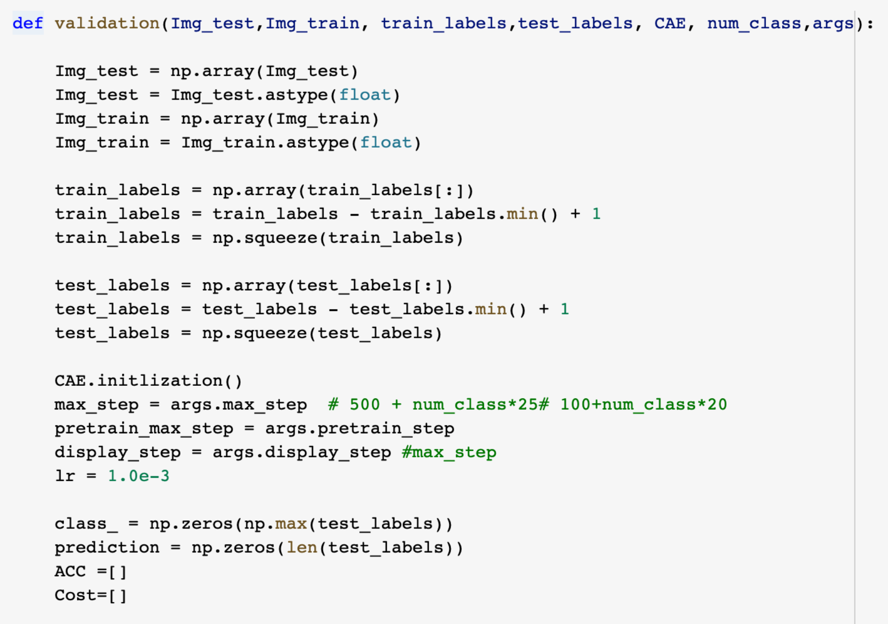
```
# Detect hardware, return appropriate distribution strategy
try:
    tpu = tf.distribute.cluster_resolver.TPUClusterResolver()
    # TPU detection

    print('Running on TPU ', tpu.master())
except ValueError:
    tpu = None

if tpu:
    tf.config.experimental_connect_to_cluster(tpu)
    tf.tpu.experimental.initialize_tpu_system(tpu)
    strategy = tf.distribute.experimental.TPUStrategy(tpu)
else:
    strategy = tf.distribute.get_strategy()
    # Default distribution strategy in Tensorflow
```

Configuring the tensor processing unit with the Google Colab notebook



```
def load_data():

    datasets = ['/kaggle/input/blood-cells/dataset2-master/dataset2-master/images/TRAIN','/kaggle/input/blood-cells/dataset2
    images = []
    labels = []

    count = 0
    for dataset in datasets:

        # iterate through folders in each dataset
        for folder in os.listdir(dataset):

            if folder in ['EOSINOPHIL']: label = 0
            elif folder in ['LYMPHOCYTE']: label = 1
            elif folder in ['MONOCYTE']: label = 2
            elif folder in ['NEUTROPHIL']: label = 3

            # iterate through each image in folder
            for file in tqdm(os.listdir(os.path.join(dataset, folder))):

                # get pathname of each image
                img_path = os.path.join(os.path.join(dataset, folder), file)

                # Open
                image = cv2.imread(img_path)
                image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

                # add padding to the image to better detect cell at the edge
                image = cv2.copyMakeBorder(image,10,10,10,10,cv2.BORDER_CONSTANT,value=[198, 203, 208])
```

Loading the blood cell dataset into the Google Colab notebook and separating the images appropriately



```
def validation(Img_test,Img_train, train_labels,test_labels, CAE, num_class,args):

    Img_test = np.array(Img_test)
    Img_test = Img_test.astype(float)
    Img_train = np.array(Img_train)
    Img_train = Img_train.astype(float)

    train_labels = np.array(train_labels[:])
    train_labels = train_labels - train_labels.min() + 1
    train_labels = np.squeeze(train_labels)

    test_labels = np.array(test_labels[:])
    test_labels = test_labels - test_labels.min() + 1
    test_labels = np.squeeze(test_labels)

    CAE.initlization()
    max_step = args.max_step  # 500 + num_class*25# 100+num_class*20
    pretrain_max_step = args.pretrain_step
    display_step = args.display_step #max_step
    lr = 1.0e-3

    class_ = np.zeros(np.max(test_labels))
    prediction = np.zeros(len(test_labels))
    ACC =[]
    Cost=[]
```

Running the sparse representation classifier model on the validation set of blood cell images

Data

Figure 1 - ROC Curves for 500 Training Images Per Cell Subtype

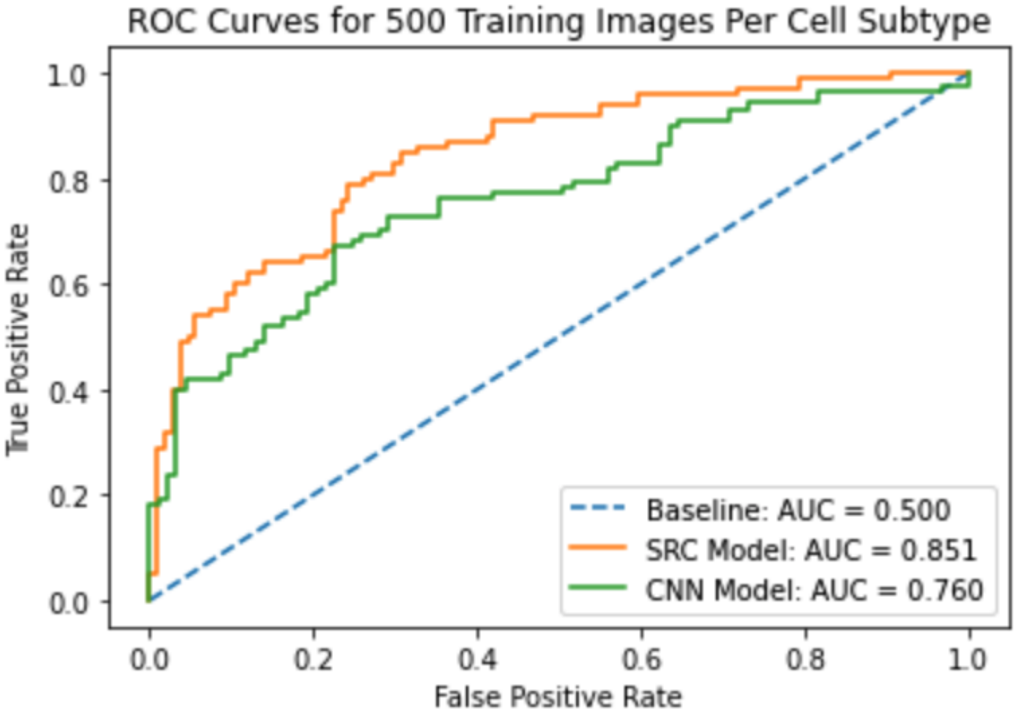


Figure 2 - ROC Curves for 1500 Training Images Per Cell Subtype

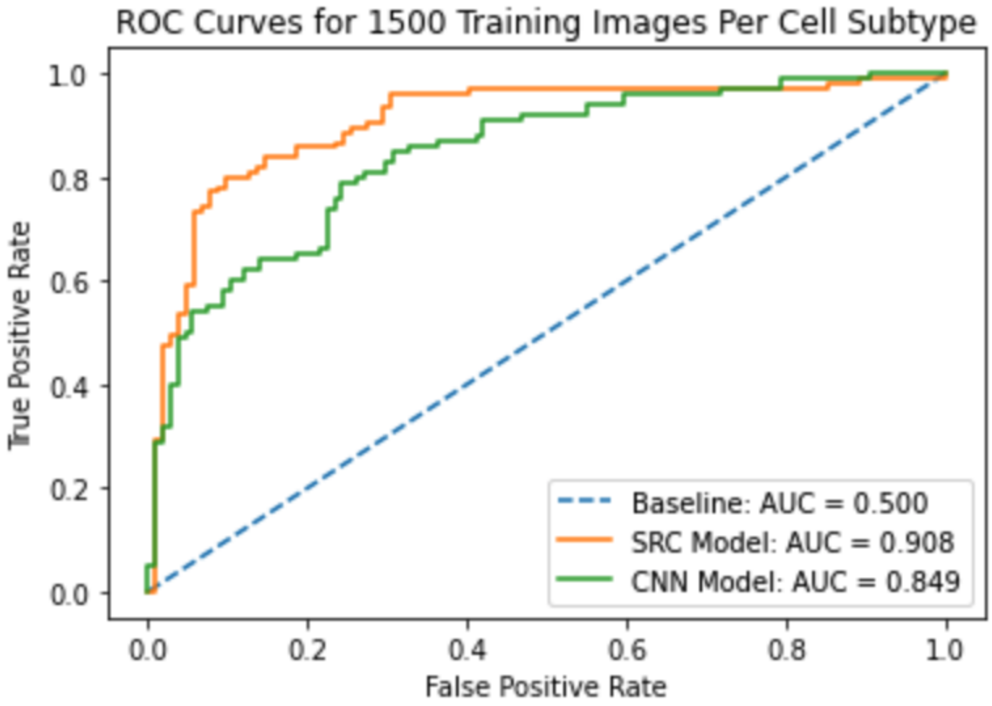
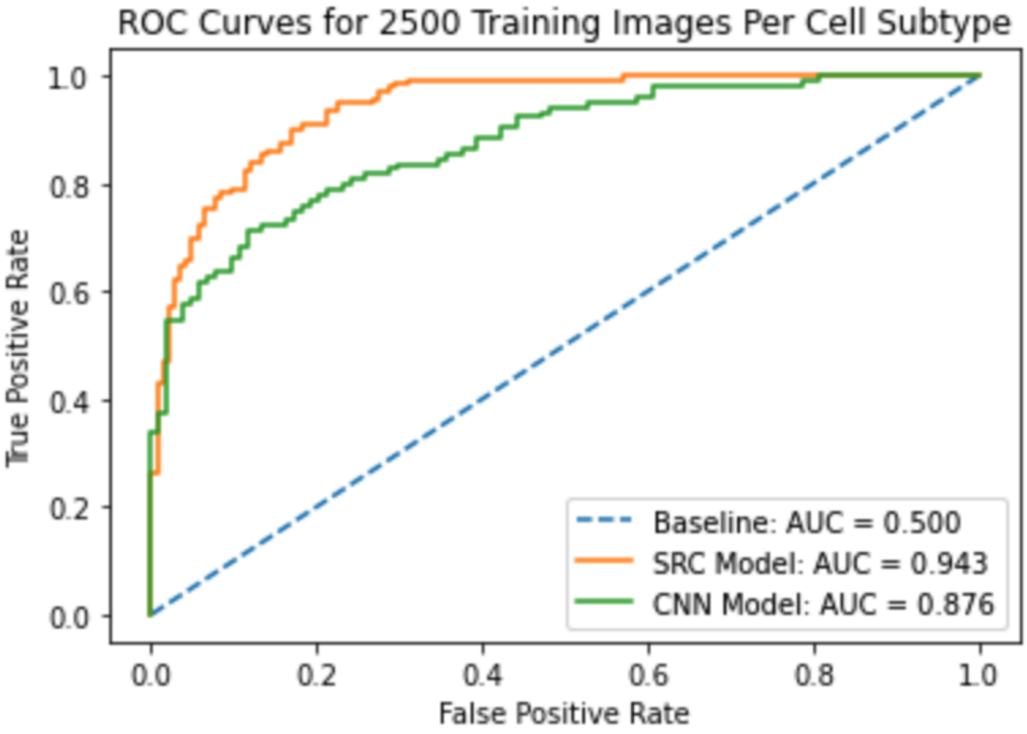


Figure 3 - ROC Curves for 2500 Training Images Per Cell Subtype



Statistical Analysis

Figure 4 - Confusion Matrix of SRC Model for Maximum Training Images

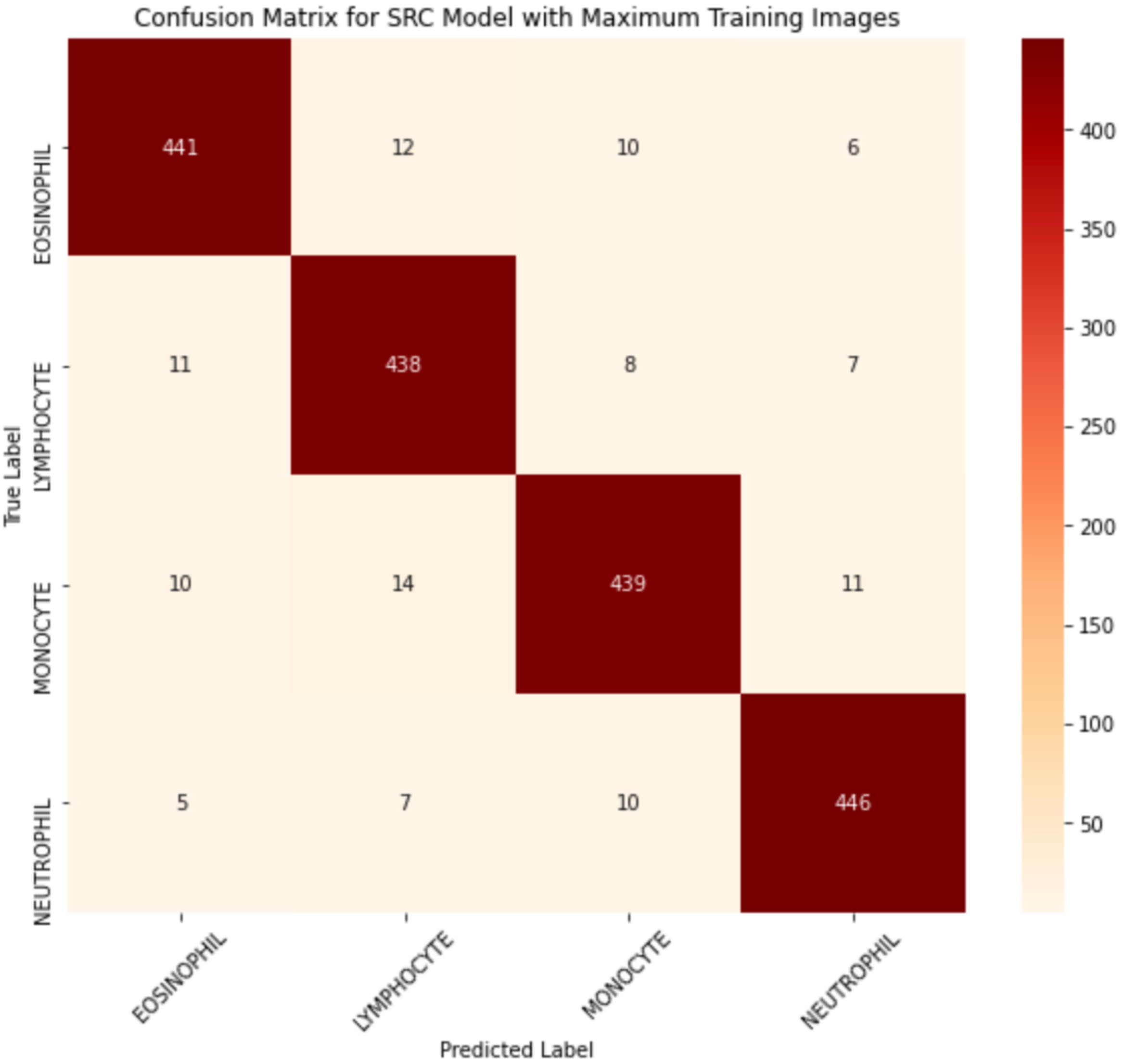
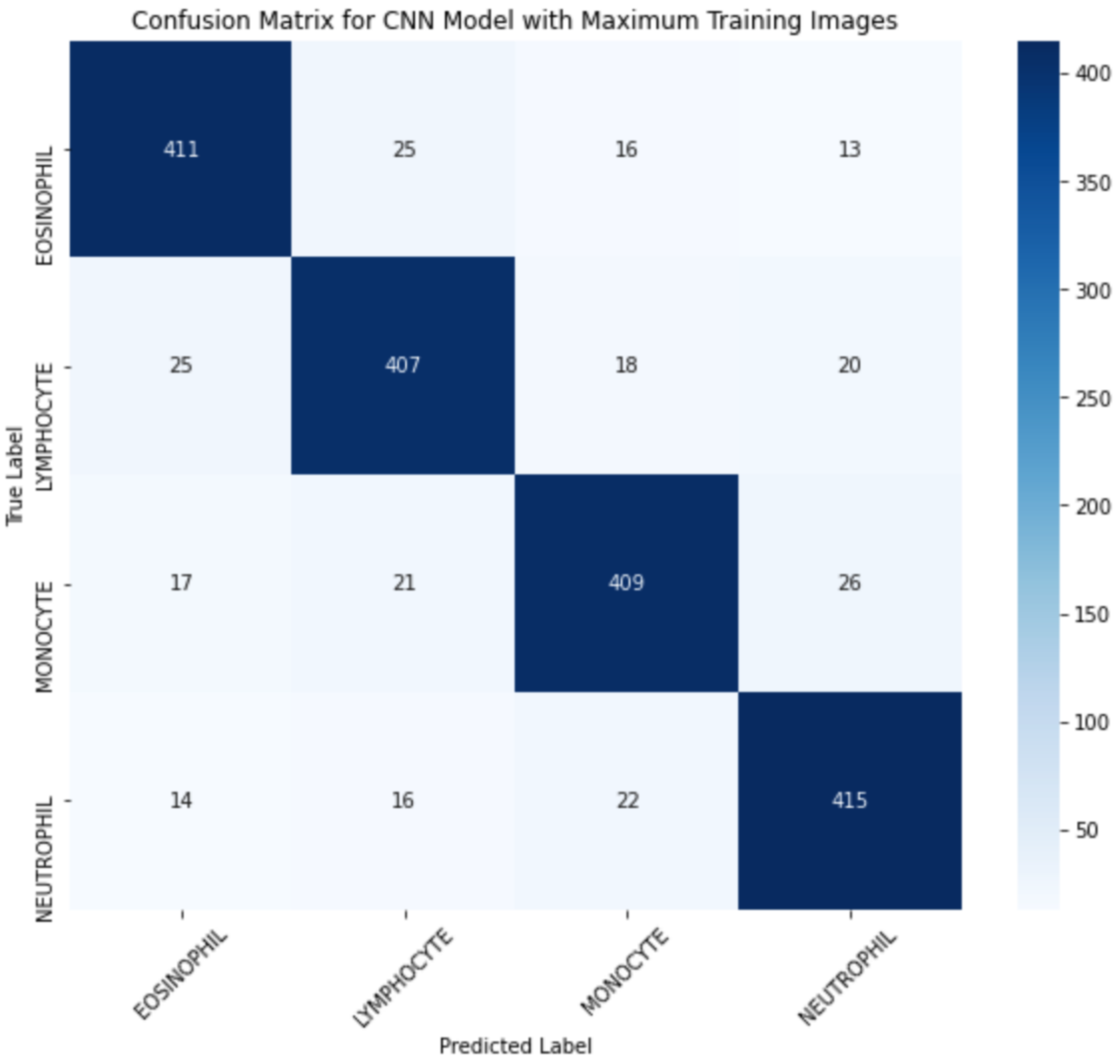


Figure 5 - Confusion Matrix of CNN Model for Maximum Training Images



Discussion

Conclusion

A sparse representation classifier (SRC) is a type of algorithm that utilizes patterns to recognize characteristics of data in a low-dimensional space, which is particularly useful for analyzing microscopy, and typically requires less data to train the model. The purpose of this experiment was to develop a sparse representation model running on a tensor processing unit that could identify and characterize patient blood samples for blood-based diseases during real-time microscopy using fewer data examples for training compared to a convolutional neural network (CNN). The results demonstrate that the sparse representation classifier can achieve high accuracy with less microscopic blood images when compared to the standard convolutional neural network. When the SRC was run for 500 training images per cell subtype, the accuracy of the model was 85.1%, which is greater than the accuracy of 84.9% for the convolutional neural network with 1500 training images per cell subtype, approximately three times the number of training images. When the sparse representation classifier was trained using 2500 training images per cell subtype, the accuracy of the model was 94.3%, which is 6.7% greater than the accuracy of 87.6% for the convolutional neural network with 2500 training images per cell subtype as shown in Figure 4. Additionally, when the sparse representation classifier was run for 500 training images per cell subtype, the accuracy of the model was 85.1%, which is 9.1% greater than the accuracy of 76.0% for the convolutional neural network with 500 training images per cell subtype as shown in Figure 2. By observing the confusion matrices of both models for the maximum training images of 2500, the number of false classifications by the sparse representation classifier is much smaller compared to the number of false classifications by the convolutional neural network as shown in Figures 5 and 6. In conclusion, the hypothesis, if a sparse representation classifier is developed running on a tensor processing unit, then the model will be able to identify and characterize white blood cell subtypes for blood-based diseases during real-time microscopy using half the number of data examples for training compared to a convolutional neural network, was supported by the data gathered from the experiment.

Applications

Using this research, doctors will be able to minimize the amount of blood data, and thus the cost, associated with using cell subtypes for blood disease diagnosis. Sparse representation classifiers can achieve high accuracy with less microscopic blood images when compared to other models, all while avoiding human error through computer aided classification. In addition, the use of a tensor processing unit reduces the overall time associated with analysis and processing of blood images, opening up the possibility of utilizing this hardware for reducing the costs of diagnosing other maladies.

Limitations

One of the limitations of this research is that the selected blood image dataset is not entirely representative of the total population of subtypes for individuals afflicted with blood-based diseases. There are many challenging diseases, such as leukemia, that have various subtypes depending upon cell morphology, so the accuracy of the model may be slightly compromised when exposed to a different sample of blood data. Another limitation is posed by lack of understanding of how exactly these algorithms are building and selecting the characteristics, or features, they use to classify data inputs.

Error Analysis

An example of random errors could include inherent defects within the white blood cell image dataset itself, loss of data during the download and query process, and corruption of portions of data during the transfer of files from directory to directory.

Future Research

Future research can be done to expand the capabilities of this algorithm for detecting other white blood cell subtypes, such as basophils and macrophages, as well as implementing this sparse representation model for real-time microscopy. Using augmented reality and cameras oriented through microscopic lenses, this algorithm can be developed to classify cell subtypes in real time.

References

Plenge, E., Klein, S.S., Niessen, W.J., Meijering, E. (2015). Multiple Sparse Representations Classification. PLOS ONE, 10(7), e0131968. https://doi.org/10.1371/journal.pone.0131968

Zhu, Q., Feng, Q., Huang, J., Zhang, D. (2016). Sparse representation classification based on difference subspace. IEEE Congress on Evolutionary Computation (CEC), pp. 4244-4249. doi: 10.1109/CEC.2016.7744329

Sadafi A. et al. (2019). Multiclass Deep Active Learning for Detecting Red Blood Cell Subtypes in Brightfield Microscopy. In: Shen D. et al. (eds) Medical Image Computing and Computer Assisted Intervention – MICCAI 2019. MICCAI 2019. Lecture Notes in Computer Science, vol 11764. Springer, Cham. https://doi.org/10.1007/978-3-030-32239-7_76

Ghane, N., Vard, A., Talebi, A., & Nematollahy, P. (2019). Classification of chronic myeloid leukemia cell subtypes based on microscopic image analysis. EXCLI journal, 18, 382–404. https://doi.org/10.17179/excli2019-1292

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature 521, 436–444. https://doi.org/10.1038/nature14539

FY22 ISEF Rulebook.pdf. (n.d.). Retrieved July 30, 2021, from https://drive.google.com/file/d/1EqHYMQYnygRsLobmzA9oyOCkrt7JmV40/view

Society for Science and the Public (2016-17). International Science and Engineering Fair 2016-17: International Rules & Guidelines. Washington, DC: Society for Science and the Public.