CS 5395 Project Report
Texas State University
Fall 2019

Mahya Saeednejad
m-s944
m-s944@txstate.edu

Abstract

This project investigates the possibilities of using machine learning algorithms as a tool for early detection of anemia in goat's blood using fluorescence spectroscopy Excitation Emission Matrix (EEM) measurements. Our model was originally developed for detecting colorectal cancer in human plasma and can be applied to anemia detection because of its ability to detect the differentiating factors between goats with and without anemia.

A total 299 Samples were collected included patients with colorectal cancer or with adenomas, and from individuals with other non malignant findings or no findings. Machine learning algorithms such as SVM and convolutional neural network applied on the data and their performance evaluated.

1. Introduction

Every year, hundreds of goats and other animals die in ranches because of anemia. By the time people can detect anemia in goats it is often too late to save their lives, which has led to a huge loss of resources.

Anemia occurs when there is not enough healthy red blood cells to carry adequate hemoglobin from the lungs to the body. This causes a lack of oxygen in the body which leads to weakness, shortness of breath, dizziness, cardiac dysrhythmia, and headache. Anemia can also be caused by blood loss from injury, internal parasites (e.g. barberpole worm or liver fluke), and external parasites (e.g. lice, ticks, etc). Anemia can be temporary or long-term, and it can range from mild to severe.

Early detection and diagnosis of anemia greatly increase the chance of successful treatment. Today, detection of anemia is most likely when an actual veterinarian visits the goats and observes the symptoms; however, by this time, the goat anemia is more severe which makes treatment more complicated and sometimes impossible.

Therefore, researches look for a way to detect small changes in the blood which help to indicate the quantity of iron in the blood in order to identify the likelihood of having, or developing, anemia. Fluorescence spectroscopy of a sample blood is one of the most inexpensive and efficient ways to detect the onset of anemia.

The idea of using auto-fluorescence measurements of blood was first developed to identify people with cancer by Leiner, Wolbeis, and fellow researchers in the 1980s. They considered the fluorescence excitation emission matrix (EEM) of a diluted blood serum sample as a base for pattern recognition to monitor the health status of a person.[2]

Their hypothesis is that due to the high sensitivity of fluorescence spectroscopy it is possible to observe the small deviations in the fluorescence spectrum and distinguish between patients with and without cancer. (Leiner

et al. 1983, 1986; Wolfbeis and Leiner 1985).[2]

In this study we explore the ability to implement machine learning as an alternative tool for detecting anemia using fluorescence spectroscopy of blood samples.

2. Problem Statement

Early detection of anemia provides better opportunities for goats to obtain more effective treatment with fewer side effects. Goats whose anemia are found early and treated in a timely manner are more likely to survive than those whose anemia are not found until the symptoms appear.

Researches look for an alternative way to detect very small changes and find a pattern in fluorescence spectroscopy data. Machine learning can help us to predict a pattern which may indicate the likelihood of a future diagnosis of anemia. Our model is structured to predict the early blood related diagnosis with the spectrometer data.

One of the biggest challenges of the this project was to get samples of goats' blood who are healthy and who have anemia. Due to the lack of access to such a database, we design the pipeline with fluorescence spectroscopy of color papers and the goal is to design a methods to classify different color papers based on their spectrometer data. Color paper data set has the same format as the samples of goats' blood. Therefore, the designed pipeline can be reused for goats' blood when the data set will be ready.

3. Color papers Data set

The dataset includes 4 color papers including red, orange, green, and blue colors. Colors measured in the spectral area with excitation of visible wavelengths from 400 to 700 with 10 nm increment, and emission wavelengths from 179.188 to 870.241 with 0.29 nm increment.

3.1. Color Data Pre-processing

We observe that all the intensity values for the 179.188 emission wavelength and few other intensity in different emission wavelength are negative numbers. Since intensity can not be negative number, the error could be result of a problem with spectrometer tools. So, in the first step of pre-processing, we delete all the intensity for 179.188 wavelength and for other negative intensities in different emission wavelengths, we replace them with the average intensity values of next and previous wavelengths' intensities.

In the next step, for each excitation wavelength, we replace intensities with zero for emission wavelengths less than excitation wavelengths. For instance, for the 400 nm excitation wavelength, we replace all intensities from 179.562 nm to 400 nm with zero.

In the last step, In order to remove the spectrometer zooming affect on the intensities values, we normalize all the intensities for each sample. For normalizing, we divide each intensities by the sum of all intensities.

All the pre-processing steps are re-usable for goats' blood samples.

3.2. Color papers Visualization

In order to determine the best machine learning method for classification of our data, first we need to understand data characteristics. Therefore, make heat-maps and plots of each data sample.

Matplotlib and seaborn, which are the widely used libraries are employed to visualize the data in this project.

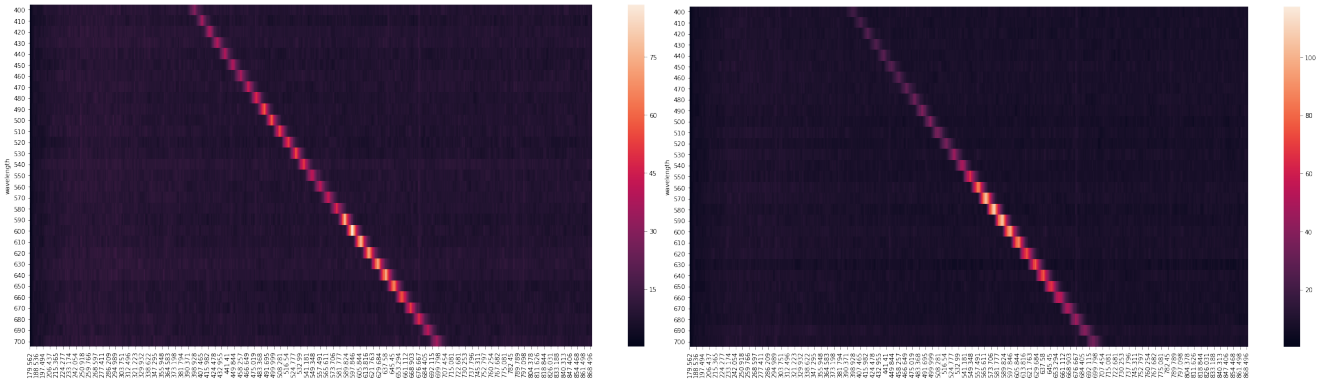


Figure 1. heatmap of red color paper after pre-processing

Figure 2. heatmap of orange color paper after pre-processing

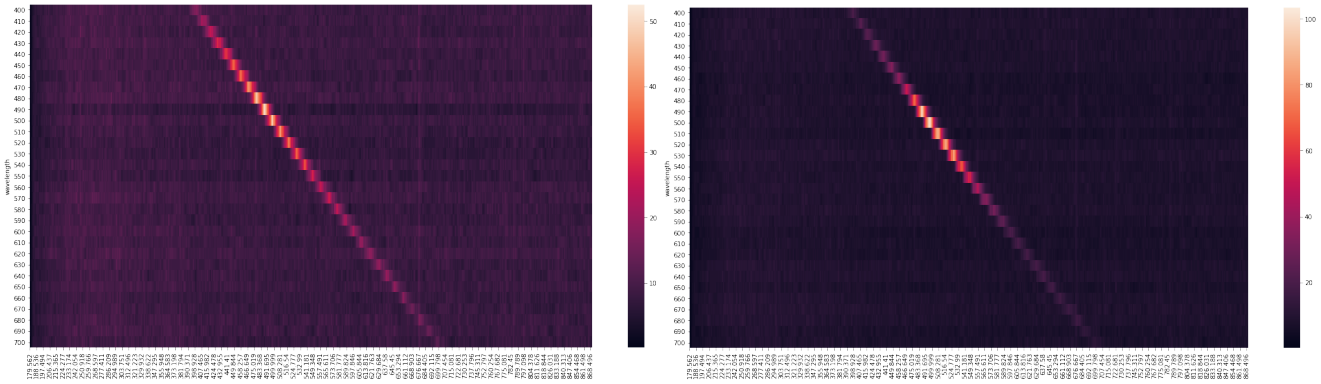


Figure 3. heatmap of blue color paper after pre-processing

Figure 4. heatmap of green color paper after pre-processing

3.3. Methods

Convolutional Neural Network(CNN) is a one of the dominant Deep Learning algorithm in many research areas-especially in image recognition, image classifications, Objects detection and face recognition.

Usually CNN includes convolution layers and pooling layers. Convolution layer is the first layer in CNN which apply learned filters to input in order to create feature maps. Convolution layers in CNN allows layers close to the input to learn low-level features such as lines, and deeper layers in the model to learn high-order or more abstract features, like shapes or specific objects.

A pooling layer is a new layer added after the convolutional layer. Specifically, after a nonlinearity such as ReLU has been applied to the feature maps. Pooling operators such as Max-pooling extract the most prominent structures. The combination of convolutional and pooling layers extracts features. Therefore, pre-processing and feature extraction required in a CNN is much lower as compared to other classification algorithms, since CNN is designed to extract features from an input with different levels of abstraction.

The fully connected layers is attached the end of the network. This layer basically takes an input volume (whatever the output is of the convolution or ReLU or pool layer preceding it) and outputs an N dimensional vector where N is the number of classes that the program wants to classify. Since all layers are trained together, CNN

integrates feature extraction with classification.

As we can see in the heat-maps, each sample has different background color and brightness in the line in the middle of the image. Therefor, we use Convolutional Neural Network(CNN) as a classification tool since it can learn colors, patterns, shape and many other characteristics of each sample, and based on the extracted feature in convolutional layers, it can classify them into different classes with fully connected layers.

Support vector machine(SVM) is another powerful algorithm in machine learning for classification. In SVM, a hyperplane is selected to best separate the input samples by their classes. Hyperplane is decision boundary that helps classify the data points in variable space.

The distance between the hyperlane and the closest data points is referred to as the margin and the closest point are referred as Support vectors. Using these support vectors, we maximize the margin of the classifier. The best or optimal hyperlane that can split input points is the hyperlane with the largest margin. In this research, the margin assumption is relaxed, it means we allow some observations to violate the margin condition in order to avoid over-fitting.

The primary models decided were SVM and CNN. SVM would be used to test the data in numeric format where as in CNN, the data would be considered as images, feature extraction will be performed and then used.

we evaluate the performance of each methods and we can choose the best method for spectrometer data classification problems. In both methods, we need to split data set into two group of training and testing set. With training set the model will learn how to classify data and with test set we evaluate the performance of the model. Usually 80% of the data is used for training and the rest 20% is used for testing.

However, since we have only one sample of each color paper, it is not possible to split data set into test and training set. Therefore, In order to apply models and evaluate their performances, we use a blood plasma public data set.

4. Plasma Data set

The public data set used for this project consists of three control groups which are (1) healthy subjects with no findings at endoscopy, (2) subjects with other, non malignant findings and (3) subjects with pathologically verified adenomas (Lomholt et al. 2009). Each of the groups, case and controls, consisted of samples from 77 individuals. Additional control samples, standardized pooled human citrate plasma, were purchased from 3H-Biomedical AB,Sweden.[2]

The dataset includes 299 samples. The undiluted spectra measured in the spectral area with excitation wavelengths from 385 to 425 with 5 nm increment, and emission wavelengths from 585 to 680 with 1 nm increment.[2] Supplementary information such as age, gender, smoking habits, and co-morbidity are also provided for each sample.

The samples were measured undiluted in Phosphate Buffered Saline (PBS) (pH 7.4). The diluted samples were prepared immediately after the samples were thawed, and then stored on wet ice (0C) until measured (app. 20 min).[2]

The undiluted samples were measured with excitation from 385 to 425 with 5 nm increment, and emission from 585 to 680 with 1 nm increment. Integration time was 0.02 s.

This spectral area consists of light in both the ultra violet and visual area. The ultra violet area is dominated by excitation and emission from the aromatic aminoacids tyrosine and tryptophan hence the fluorescence from proteins. The visual area covers among other things excitation and emission from vitamins and cofactors (for example riboflavin and NAD(P)H) (Wolfbeis and Leiner 1985).[2]

4.1. Related Work

Detect Colorectal cancers early provides better opportunities for patients to obtain more effective treatment with fewer side effects.Patients whose cancers are found early and treated in a timely manner are more likely to survive these cancers than are those whose cancers are not found until symptoms appear.

This project strives to find a faster and efficient way to discriminate cancer and non-cancer samples which can aid is faster treatment and understanding of their characteristics.

There are some works on various material identification using Near-infrared (NIR) spectrometry technique that collects the reflected light of a sampled material and deliver information on the biological composition and surface characteristics of materials.

Near infrared hyperspectral data collected with a miniaturized NIR spectrometer to identify cultivars of barley, chickpea and sorghum.Predictive multiclass models of 24 barley cultivars, 19 chickpea cultivars and 10 sorghum cultivars delivered an accuracy of 89%, 96% and 87% on hold-out sample. The Support Vector Machine (SVM) and Partial least squares discriminant analysis (PLS-DA) algorithms consistently outperformed other algorithms.[5]

Another research introduces the application of convolutional neural networks (CNNs) in the context of Raman spectroscopy for the identification of chemical species.The dataset contains 1671 different kinds of minerals, 5168 spectra in total. CNN combines preprocessing, feature extraction and classification in a single architecture which can be trained end-to-end with no manual tuning. Also, it show that CNN trained on raw spectra significantly outperformed other machine learning methods such as support vector machine(SVM).[7]

The current study has employed novel techniques and a combination of the results from various studies related to colorectal cancer where some focused on classifying cancer, modeling of fluorescence excitation and emission data and it used Fluorescence intensity calibration using the Raman scatter peak of water to eliminate errors in readings.[6][3][9][8]

4.2. Visualization Tools

To have better understanding of the data, its behaviour and characteristics we make heatmap and plot of data for each sample.

Matplotlib and seaborn, which are some of the widely used libraries are employed to visualize the data in this project.

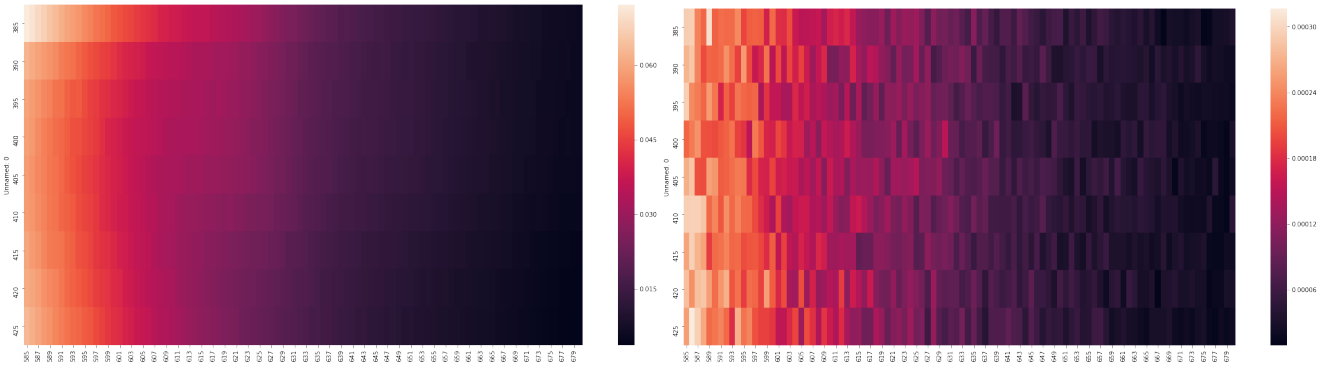


Figure 5. heatmap of non-cancer sample

Figure 6. heatmap of cancer sample

4.3. Data Processing Tools

Pre-processing methods are helpful in eliminating noise generated by spectral data. Rayleigh scatter and second order fluorescence are removed from the data and replaced with missing data and zeros using in-house software. All samples are intensity calibrated by normalizing to the integrated area of the water Raman peak of a sealed water sample measured each day prior to the measurements. This converts the intensity scale into Raman units and allows comparison of intensity of samples measured on other fluorescence spectrometers.[2]Raw spectral data were thus processed using the Savitzky-Golay and Gap-segment derivative smoothing filtering algorithms were also applied.

The spectroscopy data was provided in matlab format which was made to go through various pre-processing steps in order to get a data format that can be fed to the machine learning and deep learning models. we convert mat format file to csv file.

The excitation wavelengths, emission wavelengths, sample number and the data had to be extracted separated. All the data were then combined based on the sample number to create individual datasets. A separate samples list and labels list were generated from the given data to uniquely map the samples and their results. All the 'NULL' were eliminated to avoid errors or outliers.

wavelength	585	586	587	588
385	0.061289	0.059632	0.058012	0.055940
390	0.053567	0.052625	0.051307	0.050666
395	0.051420	0.050214	0.049273	0.048369
400	0.050779	0.049574	0.047879	0.047464
405	0.051043	0.049499	0.048293	0.047012

Figure 7. one data sample after pre-processing

4.4. Machine Learning Method

The baseline model used is SVM with linear kernal. For using SVM we need to select some features as main features which have the most information. However, since our data base is high dimensional, features are not always easy to select. Hence, we analyze performance of Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) techniques for feature extraction. PCA and LDA are both used to reduce the number of dimensions in the dataset while retaining as much information as possible.

PCA is an unsupervised learning algorithm as it ignores the class labels that maximize the variance in a data set, to find the directions. In other words, PCA is basically summarization of data. Also, PCA performs betters when variables are strongly correlated.

LDA is a supervised algorithm as it takes the class label into consideration. It is a way to reduce 'dimensionality' while at the same time preserving as much of the class discrimination information as possible.

In our research, we applied both LDA and PCA and we got better result with PCA. In order to validate the algorithm, a two-step process was performed. Original spectral data were randomly split into training and test sets.

The 28% accuracy achieved in the first attempt which shows a failure in classification. The low accuracy may be result of high dimensionality and complex data features which prevent SVM to learn features and classify the samples.

```
#using Standard Scaler
scaler = StandardScaler()
scaler.fit(X)
df_s = scaler.transform(X)

# #applying PCA
pca_model = PCA(n_components=100).fit_transform(df_s)
pca_df = pd.DataFrame(data = pca_model)
# pca_df
```

```
# Savitzky-Golay filter with second degree derivative
sg=savgol_filter(pca_df,window_length=11, polyorder=3, deriv=2, delta=1.0)
```

```
X_train, X_test, y_train, y_test = train_test_split(sg, y, train_size=0.8, random_state=23,stratify = y)
```

```
#Using Support Vector Machine
clf = svm.SVC(kernel="linear")
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)

classifier = svm.LinearSVC(random_state=23)
classifier.fit(X_train, y_train)
y_score = classifier.decision_function(X_test)


cm = confusion_matrix(y_test, y_pred)
print(cm)
print('Accuracy' + str(accuracy_score(y_test, y_pred)))
```

Figure 8. SVM implementation

4.5. CNN Method

A CNN was built by using Keras v.1.2.2 with Tensorflow v.2.0 as the computation back end. Each input sample size reform from 864 to 9*96 Excitation Emission Matrix. 9 represents nine input wavelength and 96 represents the output length wave. The input was fed to three two dimensional convolutional layers[64, 32,16] with relu activation and each layer followed by MaxPooling layer, the Pooling layers are responsible for reducing the spatial size of the Convolved Feature. The output from the convolutional kernels were flattened and a dropout layer was added before connecting to a two fully connected dense layer with a relu activation function.

The output layer was a single dense neuron with a Softmax activation function. The loss function was a categorical crossentropy. Training was done with the adam and adadelta optimizer.

The low number of accuracy can be result of limited number of data, amount and complexity of the parameters and CNN architecture. we used data augmentation and different CNN architecture to improve deep learning model performance.

We applied CNN on original data set with 299 samples and original data set added by augmented data with 1496 samples. Table 2 shows the accuracy for various experiments. In this experiment, we consider four class in the data set.

Number of layer	Optimizer	Epochs	Accuracy
3	Adadelta	100	0.3559
3	Adam	100	0.4576
3	Adam	200	0.4915
3	Adam	300	0.5932
3	Adadelta	200	0.3898
3	Adam,SVM	300	0.3220

Table 1. CNN results in various experiments

```
from sklearn import preprocessing
lb = preprocessing.LabelBinarizer()
y_train = lb.fit_transform(y_train)
y_test = lb.fit_transform(y_test)
```

```
model = Sequential()
K.set_image_data_format('channels_last')
model.add(Conv2D(64, (9, 9), padding= 'same', input_shape=(9, 96, 1), activation= 'relu' )) #9*96*30
model.add(MaxPooling2D(pool_size=(2, 2))) #4*48*20
model.add(Conv2D(32, (4, 4), activation= 'relu' )) #2*46*15
model.add(MaxPooling2D(pool_size=(1, 2))) #1*23*15
model.add(Conv2D(16, (1, 5), padding= 'same', activation= 'relu' ))
model.add(Dropout(0.1))
model.add(Flatten())
model.add(Dense(128, activation= 'relu' ))
model.add(Dense(50, activation= 'relu' ))
model.add(Dense(4, activation= 'softmax' ))

opt =optimizers.Adadelta(lr =0.1)

  # Compile model
model.compile(loss= 'categorical_crossentropy' , optimizer= 'adam' , metrics=['accuracy'])
```

```
model.fit(X_train, y_train, epochs=300, batch_size= 170)
score = model.evaluate(X_test, y_test, batch_size=170)
```

Figure 9. CNN implementation

4.6. SVM-CNN architecture

In the previous CNN architecture we used Softmax in the last layer of neural net work. Softmax is a powerful activation function that turns numbers into probabilities that sum to one. Softmax function outputs a vector that represents the probability distributions of a list of potential outcomes. Usually Softmax is used for classification problems.

The cited study introduced the usage of linear support vector machine (SVM) in an artificial neural network architecture.[10]The study showed using SVM, instead of Softmax, as a binary classifier in the last layer of CNN produces better results. The idea behind it is that CNN extract the features and SVM uses the features extracted by CNN for classification.

The other studies, show that CNN combined with SVM did not improve the performance for image classification for multinomial classification with relatively simple CNN model.[1]

As suggested in the former paper, we used SVM in the last layer of CNN. However, it did not improved the performance of CNN which may be result of multi classification or simplicity of the model,as suggested in the latter paper.

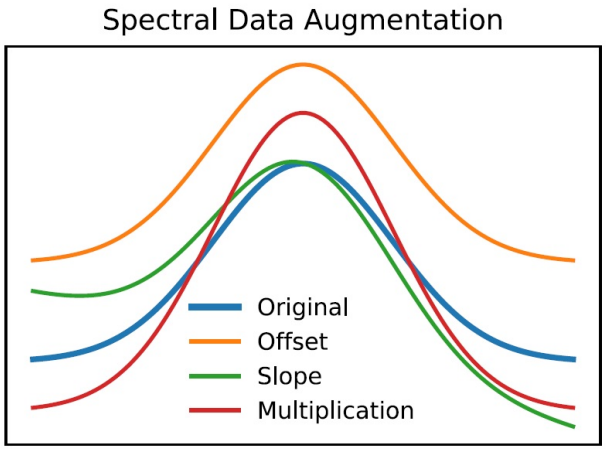
4.7. Data Augmentation

Having a large dataset is crucial for the performance of the deep learning model. When we train a CNN model, what we're really doing is tuning its parameters such that it can map a particular input to some output (a label). When our data has a lot of parameters, we would need to show our model a proportional number of examples, to get good performance. Also, the number of parameters you need is proportional to the complexity of the task your model has to perform.[4]

One of the ways of dealing with small data set is data augmentation which encompasses a suite of techniques that enhance the size of training data sets from the limited number of labelled samples by simulating various expected variations in the data sets such that better deep Learning models can be built using them.

In computer vision, there are very famous and easy to understand methods to create new images in such a way that content of the image is recognizable. flipping images horizontally and vertically, Rotating the image by angles, scaling an image outward or inward angles, cropping the image background or adding noise to the image.

For spectral data the variation employed is suggested as in Figure 2. Here random offsets, random changes in slope and random multiplications are added to the existing spectrum to expand the data set.[4]



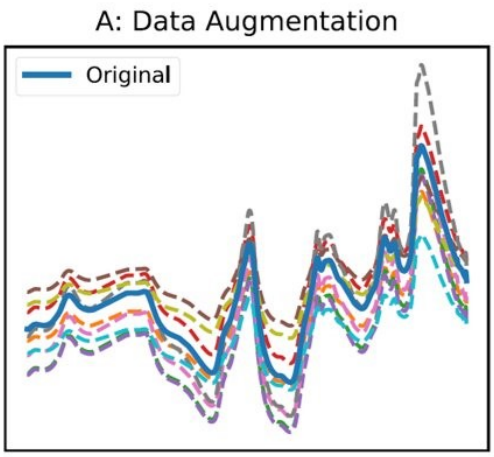


Figure 10. spectral data augmentation

Figure 11. spectral data augmentation effects

In this paper, Some of the data sets were augmented by adding random variations in offset, and multiplication. Offset was 0.10 times the standard deviation of the training set. Multiplication was done with 1.10 and 0.90 times the standard deviation of the training set. So we create 4 additional samples out of each sample in the original data set. The number of augmented data was 1197 samples.

In order to validate the augmented data and to see if they are good representative of original data, we compared the plots and heat-map of augmented data with original data and we see all the images have the same pattern with different range.

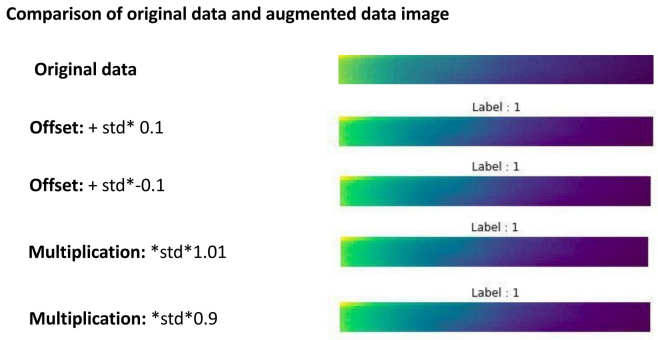


Figure 12. heatmap of augmented data and original data

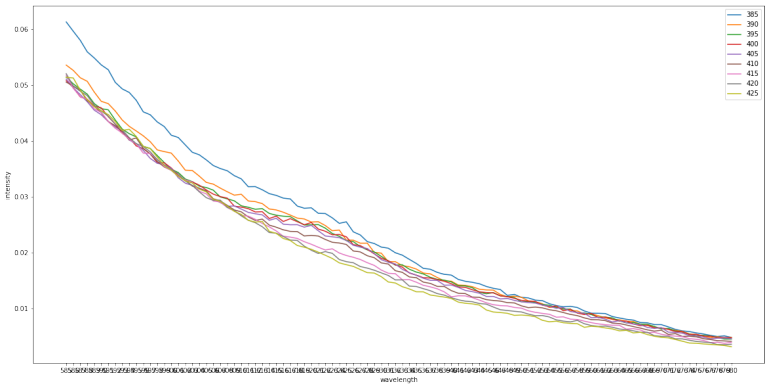


Figure 13. original data sample

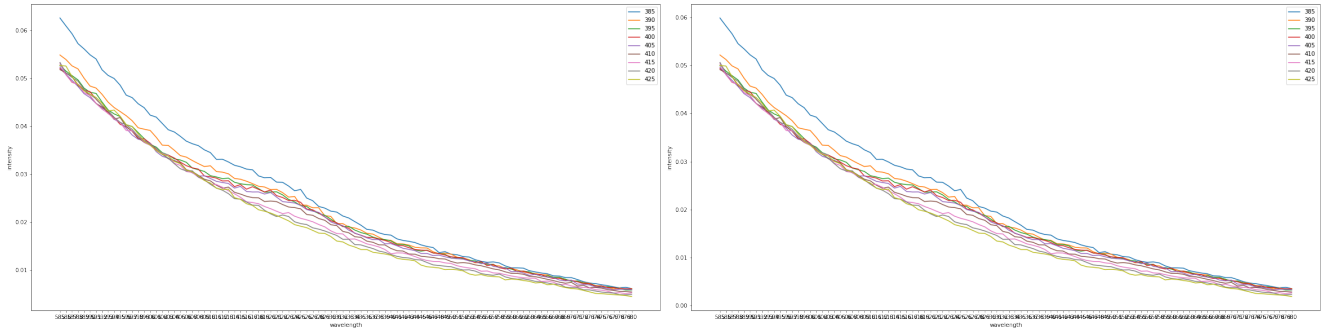


Figure 14. augmented data: Offset: + std* 0.1

Figure 15. augmented data: Offset: - std*0.1

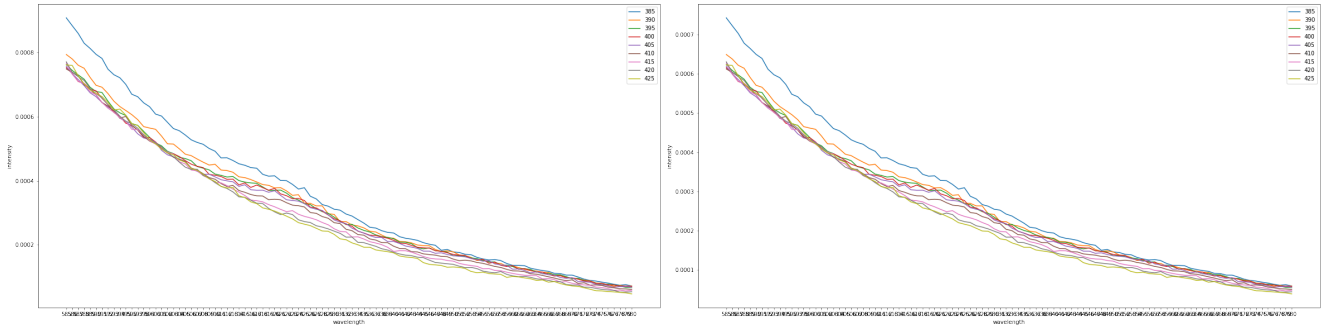


Figure 16. augmented data: Multiplication: *std*1.01

Figure 17. augmented data: Multiplication: *std*0.9

Dataset	Size
original data set	299
augmented data set	1197
total	1496

Table 2. datasets size

4.8. Results for multi-classification

In table 3, we see the results of different SVM, CNN and combination of SVM with CNN with original dataset(299 samples), only augmented data(1197 samples) and combination of original data and augmented data(1496 samples).

Model	**Accuracy**	**Number of samples**
SVM	28%	299
CNN	31%	299
SVM+CNN	31%	299
SVM	98%	1197
CNN	31%	1197
SVM+CNN	26%	1197
SVM	96%	1496
CNN	29%	1496
SVM+CNN	26%	1496

Table 3. Result of SVM, CNN and combination of CNN and SVM on three dataset with four labels

As we see in the table, non of the models perform well on classification. SVM accuracy jumps from 28% to 98% for the augmented data which is unreliable result and shows over-fitting.

4.9. Results for binary classification

Since our initial goal was to detect colorectal cancer and non-cancer groups. we combine three classes which do not have colorectal cancer into one. Then, we apply the models again on the data set and we get higher accuracy in all models compare to the classification of four groups.

Model	**Accuracy**	**Number of samples**
SVM	75%	299
CNN	86.44%	299
SVM+CNN	79.66%	299
SVM	74.91%	1197
CNN	80.13%	1197
SVM+CNN	79.92%	1197
SVM	74.91%	1496
CNN	77.21%	1496
SVM+CNN	76.41%	1496

Table 4. Result of SVM, CNN and combination of CNN and SVM on three dataset with two labels

In-contrast to the previous result which we see a big jump in accuracy, when we applied models on three different data set with two labels, the accuracy are close to each other which means models could successfully classify cancer and non-cancer groups. Also, we observe that in all three data set CNN, CNN+SVM and SVM respectively had the most accuracy.

Therefor, we can conclude that, for the goat research which has two labels for classification, our model will be able to classify them as anemic and non-anemic. Also, CNN is the best model.

5. Conclusion

Early detection of diagnosis can save many lives. In this regard, We have introduced machine learning and CNN approaches combined with excitation emission matrix fluorescence measurements on blood plasma as an alternative method to detect blood related disgnoses. With the help of fluorescence spectrocopy we can determine small changes in blood plasma and with machine learning, systems make better decisions, at a high speed and most of the times they are accurate. Using this technique is inexpensive and it can analyze large and complex data sets. It can determine if the changes are similar to the cancer pattern and notify the patience in case any unusual patter is seen for effective treatment.

Based on this research, we find CNN with 86.44% has higher acurracy than SVM with 75% accuracy and combination of CNN with SVM with 79.66% for the fluorescence data two label classification of cancer and non-cancer patients. However, all models fail to do the four labels classification of colecteral cancer,Adenoma, patients with other type of diseases and healthy.

Furthermore, the models are able to classify goats into anemic and non-anemic group and CNN will may have the better performance.

In the next step, further researches can be developed for detecting various cancers and diagnosis using fluorescence measurements of the blood with machine learning and CNN approaches.

References

[1] A. F. M. Agarap. An architecture combining convolutional neural network (cnn) and support vector machine (svm) for image classification. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[2] I. J. C. L. N. J. H. J. N. Anders Juul Lawaetz, Rasmus Bro Maja Kamstrup-Nielsen. Fluorescence spectroscopy as a potential metabonomic tool for early detection of colorectal cancer. `https://link.springer.com/article/10.1007/s11306-011-0310-7`, 2012.

[3] R. Andersen, C. M.; Bro. Practical aspects of parafac modeling of fluorescence excitation-emission data. `https://doi.org/10.1002/cem.790`, 2003.

[4] M. G. Esben Jannik Bjerrum and T. Skov. Data augmentation of spectral data for convolutional neural network (cnn) based deep chemometrics. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[5] T. W. Frederic Kosmowski. Evaluation of a miniaturized nir spectrometer for cultivar identification: The case of barley, chickpea and sorghum in ethiopia. `https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0193620`, 2018.

[6] L. A. M. F. C. J. S. Jinchao Liu, Margarita Osadchy and S. J. Gibson. Intact and cleaved forms of the urokinase receptor enhance discrimination of cancer from non-malignant conditions in patients presenting with symptoms related to colorectal cancer. `doi:10.1038/sj.bjc.6605228`, 2009.

[7] L. A. M. F. C. J. S. Jinchao Liu, Margarita Osadchy and S. J. Gibson. Deep convolutional neural networks for raman spectrum recognition : A unified solution. `https://arxiv.org/pdf/1708.09022.pdf`, 2017.

[8] C. A. Lawaetz, A. J.; Stedmon. Fluorescence intensity calibration using the raman scatter peak of water. `https://doi.org/10.1366/000370209788964548`, 2009.

[9] E. W. W. P. K. D. P. T. K. T. A. D. T. McKnight, D. M.; Boyer. Spectrofluorometric characterization of dissolved organic matter for indication of precursor organic material and aromaticity. `https://doi.org/10.4319/lo.2001.46.1.0038`, 2001.

[10] Y. Tang. Deep learning using linear support vector machines. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

A Tools and Infrastructure

The codes are written in python and tested in JupyterLabs software. Python is used because of its concise nature, easy implementation and availability of a rich pool of machine learning libraries. JupyterLabs was chosen because of it ease of use in writing and testing codes.

At the next stage of the project, we plan to use tensorflow 2.0 for deep learning. Tensorflow 2.0 is a popular framework with Keras built-in which enables faster and easy build and implementation of custom deep learning models.

At this stage, we did not encounter any trouble with the tools or features. The current project was built and tested in Windows and MAC laptops with 16 GB RAM. GPU's or virtual environments were not experimented or needed at this stage. However, in the next steps of working with complex deep learning models, these would be employed.