**Research Paper Chagas Parasite Dataset:**

**METHOD**

Step 1 : RGB Image( Each Pixel is represented by 3 colors RGB (r , g , b)

Step 2 : Convert to Gray Scale (Reduce the AMOUNT of INFORMATION in an image and less AMOUNT of INFORMATION to PROCESS ) ( Grayscale = 0.299 R + 0.587 G + 0.114 B )(It is like a Matrix Multiplication by a Scaler Value like 0.299 by R matrix , 0.587 by G and 0.114 by B ) AND then Adding these 3 Matrix)

Step 3 : Process the Data by TRAINED ADABOOST

Step 4 : The Algorithm Detects **SUBWINDOW** of Pixels containing **POSSIBLE PARASITE.**

Step 5 : **Use Green Channel** ( For EXTRACTION of **3 Features** RELATED to **number of pixels representing a high DNA content**.

Step 6: To **DISCARD** the FALSE PARASITE , **AMOUNT of DNA Content** is used.(I think it means they see HOW MANY PIXELS are Representing DNA Content.

Step 7: Use SVM as a Classification .

This last stage is implemented with a SVM and it is very important because it allows us to include, as part of the classification procedure, **a priori knowledge about the DNA of the parasites.**

**Haar Cascade for Detecting :**

And **ONE AdaBoost Classifier**

**Based on the SHAPE of the Object , we design the HAAR-CASCADE Feature such these features try to EXTRACT Such(Particular) Pattern/Shape from the Image and hence Detect the Haar Cascade.**

1. **Generic/GENERAL Haar-like features** inspired by those proposed by Viola and Jones



**A picture containing table

Description automatically generated**

1. **Haar-like features specially** designed to **capture Chagas parasite’s morphology**

**Graphical user interface

Description automatically generated with low confidence**

**A window of TARGET SIZE is MOVED over the Image , Haar Feature is computed for each moving window.**

**Faster Computation using INTEGERAL Image Concept:**

For faster computation of Haar features we use the concept of INTEGRAL IMAGE.

**Integral Image for a pixel/point in row r and column c is**

**A picture containing graphical user interface

Description automatically generated**

𝐼(𝑟’ , 𝑐’ ) is the **gray intensity value** of pixels located in row 𝑟’ and column c’

Given the same image 𝐼 and four points 𝑝1(𝑟1, 𝑐1), 𝑝2(𝑟2, 𝑐2), 𝑝3(𝑟3, 𝑐3), and 𝑝4(𝑟4, 𝑐4) as illustrated in Figure 6(a)

A picture containing graphical user interface

Description automatically generated

**Using the definition of integral image** of (4), we can compute the regions 𝑆1, 𝑆2, 𝑆3, and 𝑆4 as follows

**Sum of pixels** in subwindow **𝑆1**, denoted by 𝜎(𝑆1)

𝜎 (𝑆1) = 𝑖 𝑖 (𝑝1) = 𝑖 𝑖 (𝑟1, 𝑐1)

**For S2 Computation:**

𝜎 (𝑆2) = 𝑖 𝑖 (𝑟2, 𝑐2) − 𝑖 𝑖 (𝑟1, 𝑐1)

**For S3 computation:**

𝜎 (𝑆3) = 𝑖 𝑖 (𝑟3, 𝑐3) − 𝑖 𝑖 (𝑟1, 𝑐1)

**For S4 computation:**

(𝑆4) = 𝑖 𝑖 (𝑟4, 𝑐4) − 𝑖 𝑖 (𝑟3, 𝑐3) − 𝑖 𝑖 (𝑟2, 𝑐2) + 𝑖 𝑖 (𝑟1, 𝑐1) .

**HAAR Cascade AND Integral Features concept on Parasite and Non-Parasite Data:**

To **compute** the **integral image** of our **Chagas Haar-like features,**

**We will Segment the Haar like Feature**

* **in minimal 3 Windows for SIMPLEST ONES**
* **in minimal 5 Windows for COMPLEX ONES**

Haar-like feature template has been **divided into 5 regions** ( R1 , R2 , R3 , R4 , R5)

**Chart

Description automatically generated with medium confidence**

To **compute the feature value**, we need to calculate :

**𝜎(𝑅1) + 𝜎(𝑅2) + 𝜎(𝑅3) + 𝜎(𝑅4) − 𝜎(𝑅5)**

This **Feature Value** is used DETECT Parasite WINDOWS.

**Analysis of DNA Spot to Discard False Windows Detected**

Helps to **discard** **false parasites** and **decreases the false positive rate** of our method.

**After Parasite is DETECTED , we now FOCUS on “ANALYSIS OF ONE SPOT (DNA ) of Pixels” in Parasite Detected Window.**

**A picture containing diagram

Description automatically generated**

Figure 7: (a) Dark spot of pixels generated by an accumulation of DNA

**PLOT Image as a Surface:**

**DNA spot** has one particular **valley-like shape**

**Chart, surface chart

Description automatically generated**

As the **SHAPE of DARK SPOT (DNA )** and the **SHAPE of LOW INTENSITY Values are DIFFERENT**

**And the SHAPE** of this dark spot and the low intensity values taken by their corresponding pixels **create a pattern**

**Thus, it becomes very useful/EASY to DISTINGUISH/DISCRIMINATE Parasite and non-parasite .**

**Based on SHAPE and Color Difference b/w DNA and other Regions we DISCARD the Incorrect Windows. (**So select I assume **Shape and Color Features** should be fetched **to distinguish** Parasite and Un parasite **Windows Detected.)**

**FEATURE EXTRACTION ( to Train the SVM) :**

Stained Pixels have values **not greater than 80**, in the **green component. So, pixels have LOW INTENSITY in Green Component.**

1. **Feature 1**: From **Feature Values** we got from Integrated Image method on Haar Features , if these **Feature Values** were GREATER than **LEARNED THRESOLD** then **we consider that WINDOW for next phase of Detection.**

Given a **sub window detected** by **AdaBoost**, the “**PERCENTAGE OF PIXELS”** that have **intensities** **at most 80** was computed.

This **feature** representsthe **SIZE of stained region of parasite**, is the **first feature.**

1. **Feature 2**: The **MEAN of the Pixels** , whose value **is Less then 80** and are **from DETECTED-Window** from Adaboost Method.
2. **Feature 3**: **STANDARD DEVIATION** of the intensities of **all pixels** with **individual intensities at most 80** was **encountered in the sub window** detected by **AdaBoost.**

**Color**, **Shape**, and **Texture** features were selected to form a **38-dimensional feature vector.**

**7 different histograms** which contain **INFORMATION** regarding RED and Green Channels.

The **hue and saturation** **COMPONENT/Channel** from the **HSV image.**

The **RESULT** of the application of the **sobel operator** **horizontally** and **vertically** over the RGB image

Once the 7 histograms are computed, **5 values of each histrogram( Mean , Standard Deviation , Kurtosis , Skewness , Entropy)**  form the **first 35 features.**

**3 more features** known as **Tamura TEXTURE parameters** are computed from the **original RGB images.**

**SVM** **Method:**

**SVM** learning requires the **computation** of “**ONE FEATURE VECTOR”** for **each subwindow** that needs to be **classified as parasite or nonparasite.**

Feature Vectors are passed through such SVM for CLASSIFICATION( as Parasite or Non-Parasite)

**Why and How chose such Features?**

Previous Research papers have used such Features for feeding in SVM ( Like in past research paper in MALARIA which is **relevant study** to this disease also used these Features) thus we used them.

**Why CHOSEN Green Component ?**

The choice of the green component over the red and the blue ones was decided after a **careful visual examination(Looking for DIFFERENCE b/w Parasite and NonParasite exist in Red or Green or Blue or in ALL CHANNELS AND they found Green Channel was Distinguishing the most )**

After such visual examination, it was clear that the **green component** was **superior to the others** in terms of **Discriminative Information. For distinguishing Parasite and Non-Parasite Images.**

We applied **Data Augmentation** : like rotation by 15 degree and generated HUGE NUMBER OF IMAGES

**Cross- Validation: SVM** with parameters **𝛾 = 0.002 and 𝐶 = 0.5**. These parameters were obtained through **multiple crossvalidation** experiments using different pairs of values for the 𝛾 and 𝐶 parameters

10-fold cross-validation procedure was used for training and testing the AdaBoost and the SVM

**840 positive examples of parasites images**

**For EVERY EXPERIMENT** , we used **756 positive images** for **training**, leaving **84 positive images** for **testing**.

**RGB Image:**

A picture containing text

Description automatically generated

**Gray Scale** and **Ada-Boost CLASSIFIER**  :

A picture containing diagram

Description automatically generated

Focus on **DOT** (DNA ) **Analysis**:

A picture containing text, picture frame

Description automatically generated

**SVM Classifier** Applied (**After Feature Extraction**):

A picture containing text

Description automatically generated

We could have used Image Segmentation instead of using Haar cascade , Adaboost and Integral Image but as Image Segmentation is slower thus we prefer Haar cascade , Adaboost and Integral Image.

Image segmentation is a **time-consuming task** and its use **without parallel computation** is almost impossible for applications that require to scan many images in a short time.

**RESULTS:**

Text, table

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated with medium confidence

Table

Description automatically generated

Table

Description automatically generated

**ROC Curves:**

Chart

Description automatically generated  
Chart

Description automatically generated

**Extra Information Outside of Paper :**

**Methods To increase Accuracy:**

We can select all the INCORRECTLY CLASSIFIED Images

Then Cluster them based on Similarity ( How??) by using any similarity matrix (based on distance of each pixel or based on OVERALL SUM OF VALUES , their mean , their STANDARD DEVIATION ) .

**Step 3 :** Then we can take each cluster of images and then WE WILL FIND the DISTRIBUTION of this PARTICULAR CLUSTER.

Then we will apply step 3 for ALL the other Clusters. So, we will get **VARIOUS Distributions** (of each cluster ) and these all distributions belong to TRAINING SET DISTRIBUTION which is the Distribution from which all these examples/images are from.

After then Take SAMPLES/IMAGES from **each of the Distributions**

Then we can Train the Model based on these HARD/DIFFICULT datasets . So that Model can learn these patterns as well .

Another easy and related method would be to:

Take Incorrect Datasets

Apply augmentation techniques on them(Incorrect Datasets) or just do not apply any augmentation and feed in these difficult datasets and then TRAIN the Model on them .

But if you apply AUGMENTATION on Incorrect datasets then SEE HOW THEY LOOK and Check HOW MUCH SIMILAR they are to INCORRECT DATASETS and How DEVIATED they are from Correct Dataset.

Make sure that They are LIKE INCORRECT DATASET and FAR from (Not necessary way to far as they might be otherwise be seen as TAKEN FROM DIFFERENT DISTRIBUTION) CORRECT Dataset. Else Model will not learn much.