Chagas Disease Vectors Identification using Data Mining and Deep Learning Techniques

Presented in Partial Fulfillment of
the Requirements for the Degree of

Master of Science

with a Major in

Electrical and Computer Engineering

in the

Department of Electrical and Computer Engineering and Computer Science

University of Detroit Mercy

by

Zeinab Ghasemi

Major Professor
Dr. Shadi Bani Taan, Ph.D.

Committee
Dr. Mohammad Fanaei, Ph.D.
Dr. Mina Maleki, Ph.D.

August 2020

Abstract

Chagas Disease (CD) is a vector–borne infectious disease transmitted from animals to humans and reversely. It is caused by the parasite Trypanosoma cruzi (abbv. as T. cruzi). It is forcing an enormous social burden on public health and counts as one of the most major threats to human health. Based on WHO statistical analysis in 2019, CD affects about 7 million people and is responsible for nearly 50,000 annual mortalities around the world. Also an average of 80 million people are living in risky areas for infection in different parts of the world.

The disease has two phases of acute and chronic. Diagnosing of CD can be performed at both acute and chronic phases. It invloves analyzing clinical, epidemiological, and laboratory data. Since controlling and treating CD is easier in the early stages, detecting it in the acute phase plays an essential role in overcoming and controlling it.

There are many clinical trials dedicated to this problem, but progress in computational research (automatic identification) has been limited. Therefore, this work presents four automated CD vector identification approaches that classify several different vectors of kissing bugs with acceptable accuracy rates. Classification of different CD vectors is important because carriers of CD belong to different species classes unevenly scattered in different parts of the world. Therefore, differentiating all species of CD vectors plays an important role in designing a robust global system for automatic identification.

Three of our proposed methods are composed of preprocessing, feature extraction, feature selection, data balancing, and classification phases. The preprocessing steps are background removal, gray–scaling, and down–sizing. The Principal component analysis (PCA) algorithm is utilized for feature extraction. A correlation–based subset selection is used for feature selection. The classes are balanced by oversampled the minority classes. Finally, the employed classification techniques include Decision Tree (DT), Random Forrest (RF), and Support Vector Machine (SVM). These three methods are named "PCA+DT","PCA+RF", and "PCA+SVM". In the fourth approach, we applied two deep convolutional neural networks (CNN) on our preprocessed dataset and omitted the feature extraction and feature selection steps. Our two convolutional neural networks VGG16 and 7–layer CNN are trained using the same oversampled image dataset.

The average accuracy using 150–features dataset for Brazilian vectors is 100% for PCA+DT and PCA+RF methods; 98.20% for PCA+SVM; 88.60% for VGG16; and 97.57% for 7–layer CNN. Brazilian vectors belong to 39 species of kissing bugs with 1620 images in the utilized dataset. The average accuracy using 150–features dataset for Mexican vectors is 100% for PCA+DT and PCA+RF; 98.40% for PCA+SVM; 89.20% for VGG16; and 96.48% for 7–layer CNN. Mexican vectors belong to 12 species of kissing bugs with 410 images in the utilized dataset.

Our results are promising and outperform previously developed systems. Given that we have a small dataset, the results of tree–based algorithms (DT and RF) are better than SVM and convolutional neural networks (CNN). Upon availability of larger datasets of kissing bugs, the results of SVM and CNN are most likely to improve.

Acknowledgements

All the rights of this research belong to University of Detroit Mercy, Department of Electrical and Computer Engineering and Computer Science.

I would like to thank my Professor and Advisor, Dr. Shadi Bani Taan, who's wisdom guided me through this path. His patience and precious guidelines kept me consistently focused during different stages of the research. He is the definition of a "true mentor" to me. I also would like to thank Dr. Victor Carmona for his help at the beginning of this research by providing information about the biological aspects of the problem.

The outcome of this research is an accepted conference paper and a submitted journal paper as follows.

1. Zeinab Ghasemi, Shadi Banitaan, Ghaith Al-refai, "Automated Chagas Disease Vectors Identification using Data Mining Techniques", accepted at IEEE 20th International Conference on Electro Information Technology (EIT2020), July-August 2020.

2. Zeinab Ghasemi and Shadi Banitaan, "Chagas Disease Vectors Identification using Data Mining and Deep Learning Techniques", submitted to the journal of Ecological Informatics.

Dedication

To my Mum and Dad who stood behind my back no matter what
and to Jeffrey, the love of my life

Contributions

1. Chagas disease is an ongoing threatening parasitic disease spreading worldwide rapidly with not much research conducted on presenting automatic Chagas disease vectors identification systems for public use. In this research, we proposed four different fully automatic identification systems for overcoming this challenge.

2. Although there is a severe shortage of sufficient Chagas disease image datasets and the available datasets are rather small, we managed to achieve very promising results for this problem. Our methods outperform previously developed systems significantly.

Table of Contents

List of Tables

List of Figures

CHAPTER 1
Introduction

> "Chagas disease is caused by the T. cruzi parasite. It is mostly transmitted by a blood–sucking insect called Triatomine bug and less frequently from mother to fetus or by digestion of contaminated food or drink. About one–third of infected individuals develop chronic heart disease. It is mostly found in Central and South America, but there are now an estimated 300,000 infected people in the United States. [1]"

Chagas Disease (CD) is a vector–borne infectious disease transmitted from animals to humans and reversely. It is caused by the parasite Trypanosoma cruzi (abbv. as T. cruzi) [2]. In 1909, it was explained in detail for the first time by Carlos Chagas. He was a Brazilian biologist and medical practitioner who worked as a clinician and researcher. He found and identified the T. cruzi parasite in the blood of Brazilian railroad workers [3]. The parasite was the reason of an acute feverish illness afflicting them. He suspected that The disease is vector–borne. The Triatomine vectors (also known as "kissing bug") were suspected to be the carriers of T. cruzi parasite [2].

To see whether these bugs harbored potential pathogens, Chagas dissected them and found numerous trypanosomes in their hindgut which he named T. cruzi. The parasite name was assigned in honour of his mentor, the Brazilian medical practitioner and biologist Oswaldo Cruz (1872–1917). Some infected bugs were sent to Cruz, where they were allowed to bite marmoset monkeys. In last than a month, the monkeys got infected and many trypanosomes were detected in their blood. Soon afterward, Chagas also discovered that the parasite was infective to several other laboratory animals. Chagas was sure that he had found a pathogenic organism of human infectious disease but did not know what kind of sickness it was.

The breakthrough came in 1909 once he was reffered to take a look at a two–year–old female patient who was feverish with enlarged spleen and liver and swollen lymph nodes. Upon first examination, no parasites were found, but four days later numerous trypanosomes were spotted in her blood with similar morphology to those

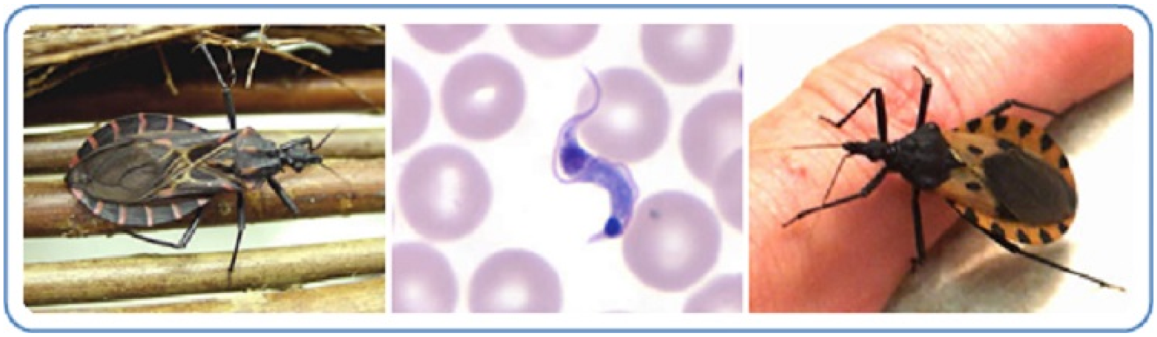


Figure 1.1: From left: the first and third images: different types of kissing bugs (if they are infected they can transmit T. cruzi). The second image: T. cruzi parasite in a thin blood smear stained with Giemsa.

previously detected in infected marmoset monkeys. Chagas had discovered a new human disease which soon bore his name [4].

The T. cruzi parasite is usually transmitted via the feces of kissing bugs, with *Triatoma infestans*, *Rhodnius proxilus*, and *Panstrongylus megistus* being the most important vectors. As T. cruzi cannot penetrate intact skin, it enters the human body through microlesions that have been introduced and contaminated with feces when individuals (mammalian host) scratch the itching vector's bite [2, 4].

Carlos Chagas described nearly all the salient features of the T. cruzi life cycle. T. cruzi is a kinetoplastid protozoan that infects vertebrate and invertebrate hosts during defined stages in its life cycle [2]. Images of various species of Triatomine bugs and T. cruzi parasite are depicted in Figure 1.1 [5].

Based on disease burden estimates of World Health Organization (WHO), CD is first among parasitic diseases in Latin America [6]. Formerly transmission was concentrated in rural areas of Latin America where poor household conditions helped vector infestation. But in the last few decades, due to successful vector control programs, transmission in rural areas is decreased. On the other hand, immigration has brought infected people from rural to urban areas both inside and outside Latin America [7]. Therefore nowadays CD is also a concern for non–endemic areas in different continents such as America (Latin, Central, and North America), Europe (mainly Spain), Australia, and Asia (mainly Japan) [8]. Based on WHO statistical analysis in 2019, CD affects about 7 million people and is responsible for nearly 50,000

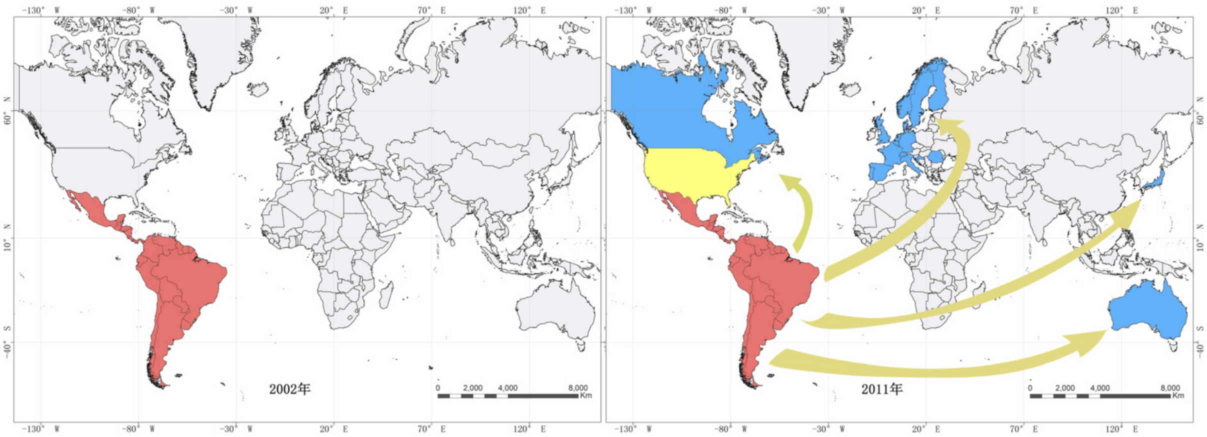


Figure 1.2: Mapping data showing epidemiological changes of CD between 2002 and 2011.

annual mortalities around the world. Also an average of 80 million people are living in risky areas for infection in different parts of the world [9, 10, 11]].

In 2007, management and control efforts in Latin America were formally joined to deal with the globalization of CD [6]. They recognized the increasing presence of imported cases in non–endemic countries including USA and the potential for local transmission through non–vectorial routes (Figure 1.3 [12]). Geospatial data from 2002 to 2011 supported this claim and demonstrated that CD exists in countries outside of Latin America, including the USA (Figure 1.2) [13]. In the Figure 1.2, red refers to endemic areas where transmission is vector–borne. Yellow refers to endemic areas where transmission is occasionally vector–borne. Blue refers to non–endemic areas where transmission is blood–borne or organ transplantation, etc. [13].

Several types of research have documented human CD in the United States. The first notifiable cases of CD was documented in 2013 and 2014 in Texas. The total of 39 human cases were reported including 12 locally acquired and 27 endemic cases [14]. The southern side of states from California to Georgia contains established cycles of T. cruzi involving several kissing bug species and different mammalian hosts such as dogs, possums and raccoons [15, 16]. But mostly, T. cruzi infected cases in the United States are Latin American immigrants moved from endemic areas [17].

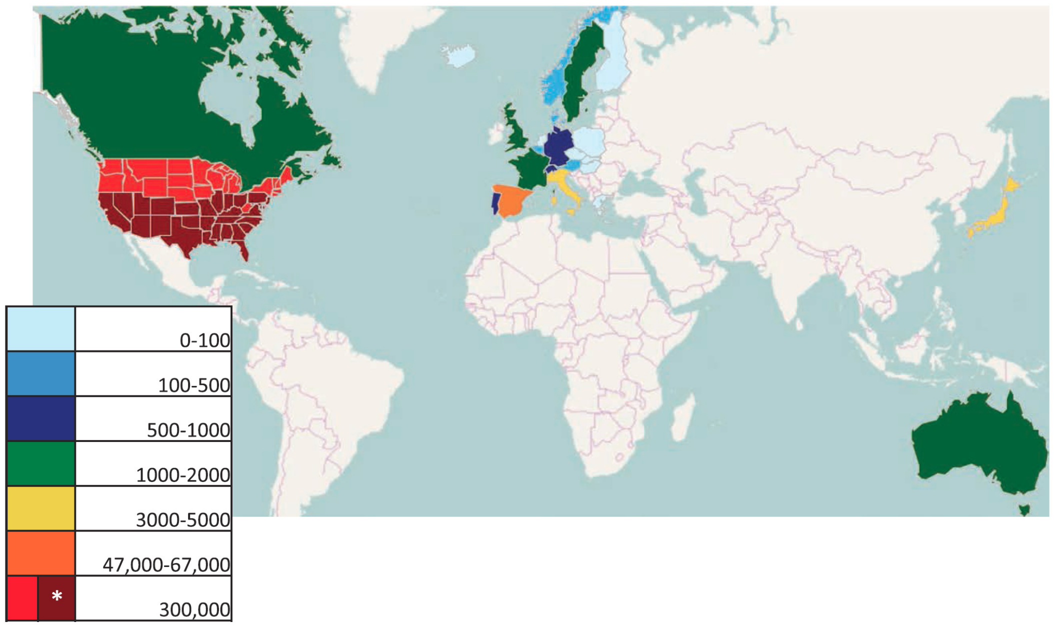


Figure 1.3: Estimated number of CD cases outside of Latin America.

1.1 LIFE CYCLE AND CLINICAL FORMS

The kissing bugs feed on blood during all the stages of their lives. They may get the T. cruzi parasite from feeding on an infected mammalian host [18]. The T. cruzi parasite is placed in the digestive system of kissing bugs. They pass on the T. cruzi parasite to other mammalian hosts through their contaminated feces placed on the bite site [19].

The CD infection has two phases. After the initial T. cruzi infection and an incubation period of 5–40 days, around 20% of infected individuals enter the acute phase. The acute phase has symptoms like high–grade fever, anorexia, abdominal pain, and local swelling around the bite site (mostly eyes and lips) [20]. Because of the mild nature of acute CD symptoms, new infections often go unrecognized [21]. The mortality rate of the acute CD is around 8%. The most number of deaths occur in young children [22]. Though the mortality rate in the acute phase is low, 70–90% of those infected become asymptomatic carriers of the parasite [21].

After a period of 4–8 weeks, the acute phase symptoms decreases and the clinical manifestations disappear spontaneously in around 90% of the cases. In this stage, the disease enters the chronic phase [23]. During the chronic phase, the infection remains clinically silent for life in around 65% of cases [19].

However, most T. cruzi infected people become asymptomatic carriers of the parasite, usually with low or undetectable parasitemia. Although T. cruzi specific antibodies and DNA may remain at detectable levels in the blood [24, 25]. But the rest of the patients develop clinical manifestations such as cardio digestive problems [26]. The chronic CD is considered a disabling disease. It is responsible for the most significant morbidity and mortality among all parasitic diseases [27].

1.2 TRANSMISSION ROUTES

Vertical transmission (vector–borne) is the most common way of T. cruzi tranmission by kissing bugs in Latin America. The kissing bugs are nocturnal feeders that live in a variety of environments surrounding human housings, including cracks and holes within the walls, ceilings and floors of housing structures.

After taking a blood meal, infected kissing bug often passes T. cruzi onto its host by excreting contaminated feces at the bite site. The T. cruzi can enter the bite wound or a nearby mucosal membrane such as the conjunctiva when the victim inadvertently rubs these parasites across their skin. Other routes of transmission include congenital, transfusion of contaminated blood, transplantation of organs from infected donors, ingestion of contaminated food or drinks, and accidental exposure (e.g., laboratory accidents). Once in the bloodstream of a mammalian host, T. cruzi can infect a variety of cell types in the body and establish a chronic infection [27]. In non–endemic areas, the transmission occurs mainly through blood transfusion, organ transplantation, or vertical transmission from mother to child [26].

1.2.1 *Vector–borne transmission*

The vector–borne transmission route, occurring exclusively within the Americas (Central and South), continues to be the predominant mechanism for new human infections. The vectorial route is considered the classic mode of T. cruzi transmission. The excretion of infected bugs contain metacyclic trypomastigotes that may enter the human body through the bite wound, intact conjunctiva (if eye is bitten) or other mucous membranes [2]. Vertorial transmission is the most interesting type of transmission from an epidemiological point of view, due to its direct connection to social, cultural, and economic aspects of a population.

1.2.2 *Blood–borne transmission*

Transfusional T. cruzi transmission was postulated in 1936 and initially documented in 1952. In 1991, the spread of T. cruzi infection in donated blood units ranged from 1 to 60% in Latin American cities. Since then, blood donation screening has become accepted as a vital pillar of the CD management initiatives [28]. Transmission of T. cruzi using blood transfusion remains probably the second most frequent transmission mechanism. This issue used to be solely evident in Latin America. Yet with the rise in immigration of CD patients to non–endemic countries, a new global scenery for this mechanism of transmission has emerged [29].

1.2.3 *Congenital transmission*

Between 1 to 10% of infants of T. cruzi infected mothers are born with congenital CD [30]. Congenital transmission may happen in a chain manner in the absence of the vector, perpetuating the disease from congenitally infected womens to their infants. The reported factors that increase the risk of this form of transmission include higher maternal parasitemia level, younger maternal age, less robust anti–T. cruzi immune responses, HIV, and parasite strain in an animal model [2].

1.2.4 *Organ–derived transmission*

The recipients of an organ from a T. cruzi infected organ donor might develop acute T. cruzi infection. However, this form of transmission is not universal. The risk from heart transplantation is thought to be higher than that from kidney or liver transplantation [31].

1.2.5 *Oral transmission*

In the recent years, rising attention has focused on the oral route of T. cruzi parasite transmission. Several outbreaks attributed to contaminated fruit or drinks have been reported from Brazil and Venezuela. Most outbreaks are rather small, affecting family groups in the Amazon region. To date, the largest reported outbreak led to more than 100 infections among students and staff at a school in Caracas. Locally prepared guava juice was implicated [32]. The oral transmission of T. cruzi is the most common way among animals in the wild cycle. Because several species of wild mammals such as small primates frequently ingest insects [29].

1.3 IMPORTANCE OF EARLY DIAGNOSIS AND TREATMENT

Diagnosing of CD can be performed at both acute and chronic phases. It invloves analyzing clinical, epidemiological, and laboratory data. In the acute phase, parasitological test can be performed to determine the presence of parasites in the blood. These tests can be direct as blood smear or thick blood smear or by multiplication as hemoculture, xenodiagnoses, and polymerase chain reaction (PCR)[1]. Due to the higher chance of curing CD in the acute phase, early diagnosis plays an essential role in the treatment process. In the chronic phase, the minimum of two serologic tests should be performed to find anti–T. cruzi antibodies. These tests include indirect immunofluorescence, hemagglutination, and enzyme linked immunosorbent assay (known as ELISA) [26]. The challenges in control, diagnosis, treatment, and clinical management at different phases of CD is illustrated in Figure 1.4 [35].

1.3.1 *Known Drugs*

The known medicine currently in use as antiparasitic therapy and proven effective for CD are Benznidazole (BZ) and Nifurtimox (NFX). The NFX was the first drug used for CD treatment in 1952. The BZ got introduced at the end of the 1970s. Nevertheless, these drugs are useful in acute cases, in congenital cases, and reactivation due to immunosuppression. However, treatment is often discontinued due to different diverse side effects [7] with a more intense side effect for NFX. Although there is

[1] PCR is a laboratory technique initially developed to make millions of copies of a particular section of DNA. It has been widely used for the diagnosis and monitoring of disease progression and therapy outcome in many infectious diseases. Since 1989, PCR strategies have been developed, aiming to analyze clinical samples infected with T. cruzi [33, 34].

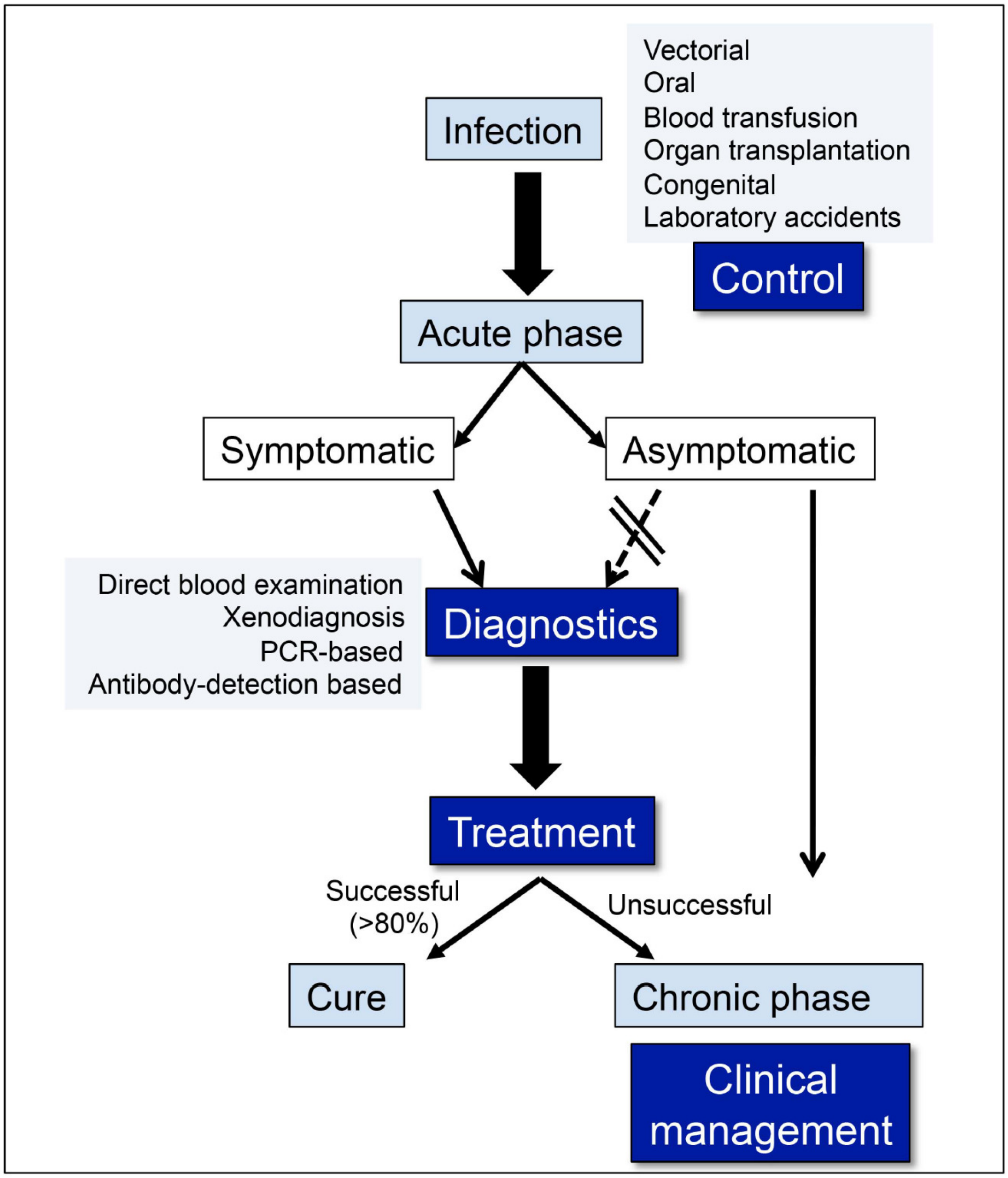


Figure 1.4: Challenges in human CD: control, diagnosis, treatment, and clinical management.

no evidence to support the use of medicine treatment in the chronic phase, some researches have demonstrated the effect of these drugs in delaying the progression of CD in the evaluated cases [26].

Regardless of the route of infection, management of T. cruzi transmission continues to be a challenge, especially considering disease emergence and reemergence. It is also critical to detect infection early to provide immediate treatment to the patients. Based on observations, it is estimated that treatment is effective in minimum of 80% of treated acute patients. Lack of detection of the acute phase or failure in treatment lead to disease chronification. Given that approximately 30% of the patients in the chronic phase will develop severe clinical forms of CD, which often results in death, clinical management is critical. However, as long as the mechanisms responsible for patient progression from the indeterminate forms to the symptomatic forms of CD are not thoroughly understood, clinical management presents another essential challenge. The search for prognostic markers of disease progression is a critical aspect for preventing pathology and introducing higher clinical measures [35].

The rest of the thesis is organized as follows. In chapter 2, the clinical trials and computational studies of CD are discussed. The background reading regarding utilized algorithms and methods in this research are explained in chapter 3. Our proposed frameworks are explained in chapter 4. The acquired results are discussed in chapter 5. Finally, this research is concluded in chapter 6.

CHAPTER 2
Literature Review

Community–based vector supervision has been common for several decades as an approach to manage CD in endemic areas (Latin America) and also recently in non–endemic areas including southern states of USA [36]. Several clinical trials and social programs are performed to raise awareness and promote health information regarding CD. They also aim to train doctors, nurses and nursing assistants, healthcare professionals, and laboratory technicians to be able to diagnose and treat CD in its initial phase [29].

2.1 CLINICAL TRIALS

In 2009, Kjos et al. [16] integrated data from multiple sources such as newly collected kissing bug species from new field studies, evaluation of preserved kissing bugs, analysis of government reports, and abstraction of peer–reviewed scientific journal articles to create a biogeographical profile of triatomine vector species found in state of Texas. The goal of their study was to assemble knowledge on triatomine bugs regarding species identification, collection site attributes, and T. cruzi infection status from diverse sources to provide a comprehensive geospatial description of endemic vector species in Texas. Their study provided new information on the distribution and infection prevalence of triatomine species in Texas. They concluded that CD vectors in Texas are widely distributed and have adapted to ecologically numerous settings. The high T. cruzi infection spread among different species surrounding human environments suggests an active peridomestic CD transmission cycle in Texas.

In 2010, a case–controlled, cohort–nested, epidemiologica study [32] was conducted throughout an outbreak of acute CD that established at a school community in Venezuela. Even though vector transmission is the prominent and classic mode of CD transmission particularly in rural areas, this outbreak was unique because it affected a large urban young middle–class healthy population and resulted in a striking public health emergency. The study was conducted to assess the extent of the outbreak and to identify possible sources of infection. Because the outbreak happened in an urban area of the city with no current vertical transmission, oral transmission (food–borne) was persumed to be the route. The rapid detection and treatment avoided high morbidity and mortality. The population of study consisted of all students, teachers, workers from the school, external persons involved with the preparation or transportation of food consumed in the school, and any person considered to be a "school contact" potentially at risk. They used PCR to evaluate a representative number of one hundred fifty blood samples and extracted the corresponding DNAs. The infection was confirmed in 103 of 1000 exposed cases and BZ and NFX was prescribed and used for severe acute phase cases. Based on their statistical analysis determined by oarasitological methods and PCR, 44 (40.7%) cases had positive test results. This rate was amongst the highest rates of parasitemia ever documented in an orally transmitted CD outbreak by that time.

In 2010, Sarkar et al. [37] assessed the spatial relative risk of the establishment of autochthonous CD cycles in Texas. Their analysis consisted of five stages of risk assessment for three most common Triatomine bugs in Texas: (1) an ecological risk analysis using predicted vector distributions. (2) an incidence–based risk analysis based on parasite occurrence. (3) a joint analysis of ecology and incidence using formal multi–criteria analysis. (4) a joint analysis using a composite risk model. Finally (5) a computation of the relative expected exposure rate taking into account human population. Their complete analysis was to argue that there is sufficient widespread risk for CD in Texas to warrant it to be declared reportable and other measures be taken. Based on their analysis, they suggested four recommendations: (1) They recommended that CD be designated as reportable in Texas as it has been in Arizona since 2007 and Massachusetts since 2008. (2) The serological status of human and canine populations should be investigated in Texas. (3) In order to prevent the establishment and spread of CD, wild species, especially rodents, merit investigation and monitoring in high–risk areas. and (4) The testing of blood donors for antibodies to CD should be made mandatory at least in high–risk areas.

In 2011, Schijman et al. [25] launched a global collaborative study by twenty six professional PCR laboratories from sixteen different countries in America and Europe. This study was the primary crucial step aiming at the analysis and validation of currently used PCR procedures for the detection of T. cruzi infection in human blood samples and towards the assessment of a regular standard operative procedure.

In 2013, Lee et al. [38] designed a Markov model to estimate the global and regional health and economic burden of CD from a societal perspective. Their Markov model structure consisted of five different five states. Their structure had a one year cycle length. The five phases are the acute disease, intermediate disease, chronic disease (cardiomyopathy with or without congestive heart failure), megaviscera, and death. Major model parameter inputs, including the annual probabilities of transitioning from one state to another, and present case estimates for CD came from various sources, including WHO and other epidemiological and disease–surveillance based reports. Using their simulation model, they estimated the economic burden of CD in comparison to other prominent diseases globally. They calculated annual and lifetime health–care costs and disability–adjusted life–years (DALYs) for individuals, countries, and regions. The concluded that productivity losses resulting from premature death are an economic argument for paying more considerable attention to CD.

In 2015, Cutris–Robles et al. [18] designed a citizen science program for CD research in 2013 and 2014 in Texas. They asked citizens to submit photos of captured kissing bugs along with corresponding information. They received nearly 4000 emails that resulted in a total of 1,980 kissing bug submissions. Over 99% of submissions were from Texas (1,968 kissing bugs), although they also received kissing bugs from Arizona, Florida, Louisiana, Oklahoma, and Virginia. In their laboratory, they identified the bugs species, measures and sex, and dissected them. They followed with DNA extraction to test them for infection. The DNA extraction was performed through amplifying T. cruzi satellite DNA quantitative real–time PCR technique for T. cruzi. They gatherd total of 1980 kissing bugs photos and they identified infection in 493 vectors. Their program provides resources for people searching for information about CD and kissing bugs in USA. They are also requesting kissing bugs samples through a variety of media including printed pamphlets, phone communication, and educational website (`http://kissingbug.tamu.edu/`), solicitations on news stations, and a dedicated web address (`kissingbug@cvm.tamu.edu/`).

In 2015, Peterson et al. [39] designed a simple mathematical model to simulate domestic vectorial transmission of T. cruzi. They specifically aimed to examine the interaction between the effects of vector management and control and the presence of synanthropic animals. They used their model to explore how the interactions between triatomine bugs, humans and animals impact the number and proportion of T. cruzi infected individuals (insects and humans). They examined the alternation of T. cruzi dynamics once control measures targeting vector abundance are defined into the system. Based on their results, in domestic T. cruzi transmission scenarios where no vector control measures are applied, a reduction in synanthropic animals may decrease the T. cruzi transmission to humans, but it would not thoroughly eliminate the transmission.

In 2018, Cucunuba et al. [40] extended their previous dynamic transmission model [39] to simulate a domestic CD transmission cycle. They examined the role of etiological treatment on CD tranmission dynamics and its potential for helping in interrupting vectorial transmission and having all infected people under care. They expanded their previous deterministic mathematical T. cruzi infection model including vector, human, and animal host populations to explore the impact of etiological treatment combined with vector control on intrrupting the transmission of CD. They utilized their previous dynamic model to compare time to intradomiciliary interruption of T. cruzi transmission in two scenarios: (1) deploying vector control (IRS) alone and (2) combining vector control with etiological treatment (measured as the proportion of parasitological cure (PPC) in the infected population). Their model suggested that control programs would benefit from combining vector control with etiological treatment of infected individuals. However, vector control's effectiveness will depend on the regional and local vector species involved in or contributing to transmission and their intrinsic susceptibility to IRS interventions. In terms of etiological treatment, model outputs show that even moderate proportions of annual PPC (10%–20%) would reduce time frameworks for achieving serological thresholds indicative of transmission interruption, infection prevalence in vectors, humans, and reservoirs, and ultimately CD burden.

In 2018, Orantes et al. [41], developed a Restriction–site Associated DNA sequencing (RADseq) [42] based pipeline for analysis of mixed species DNA extracted from Triatoma dimidiata (T. dimidiata – main CD vector in Central America) abdominal DNA. They claimed that the DNA recovered from a T. dimidiata abdomen represents a mix of DNA from the parasite T. cruzi (if present), the insect vector, possibly one or more vertebrate blood meals, and the microbial community existing within the gut, internal tissues and on the cuticle. They designed a custom bioinformatics pipeline to separate these DNA sources and analyzed them individually for either SNP genotypes [2] (T. dimidiata, T. cruzi) or taxonomic identification (blood meal, microbes). To evaluate the effectiveness of their method across a outlined spatial range (from within village dispersal to broad biogeographic and ecological differentiation), they applied a nested spatial sampling design for T. dimidiata. They started with multiple insects in individual villages and extended to samples collected from increasingly greater distances across major biogeographic regions in Central America from 1999 to 2013. They claimed their method can effectively separate genomic information of parasite, vector, microbiome and blood meal even without a sequenced genome for T. dimidiata. Their results also showed that a mixed DNA approach can provide simultaneous information about the community of biotic factors involved in T. cruzi transmission.

2.2 COMPUTATIONAL TRIALS

Clinical approaches are fairly expensive in terms of resources and expert personnel. They also need a long period of time to analyze their findings. The alternative way which is faster and therefore more effective and does not need the expertise to be performed is developing automated methods for identification of CD vectors. Therefore designing an automated system to identify CD vectors is essential and holds a great value.

[2] SNP genotyping is the measurement of genetic variations of single nucleotide polymorphisms (SNPs) between members of a species

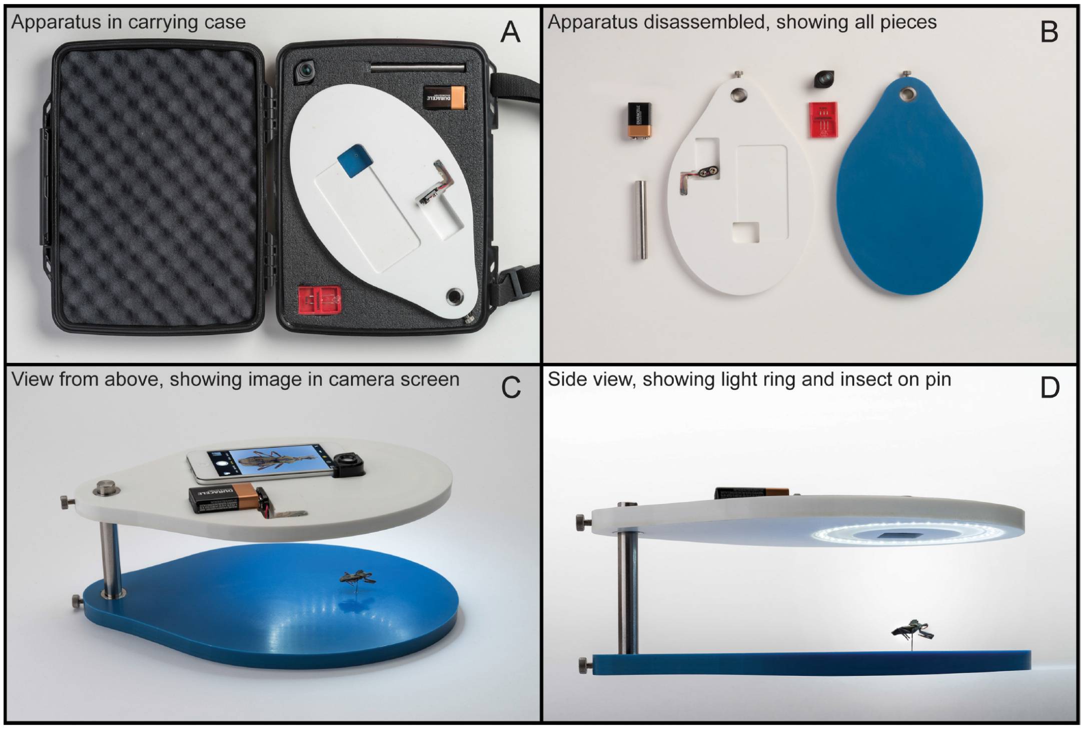

Apparatus in carrying case
A
Apparatus disassembled, showing all pieces
B
View from above, showing image in camera screen
C
Side view, showing light ring and insect on pin
D

Figure 2.1: Photographs of the device developed for capturing high–quality images of kissing bugs. (A) and (B) top view of the device; (C) and (D) lateral view of the device with the lighting ring and an insect (not a triatomine) placed on the pin.

Automation process is a two stage process. Distinguishing kissing bugs from other type of bugs and then classifying different types of kissing bugs. Both automation stages need large datasets to be feasible and there is shortage of suffiecient datasets in both fields. The current available datasets are rather small with low quality. Due to this dilemma, in this research we focus on the second step of process. This shortcoming will be addressed in future work by gathering more images for both stages.

Identifying different types of kissing bugs is essential because carriers of CD belong to several different species which are unevenly scattered in different parts of the world. Therefore differentiating all species of CD vectors plays a important role in designing a robust universal system for automatic identification of these bugs.

To the best of our knowledge not much research is carried out in this direction. There is an active research team focused on automatic identification of CD vectors

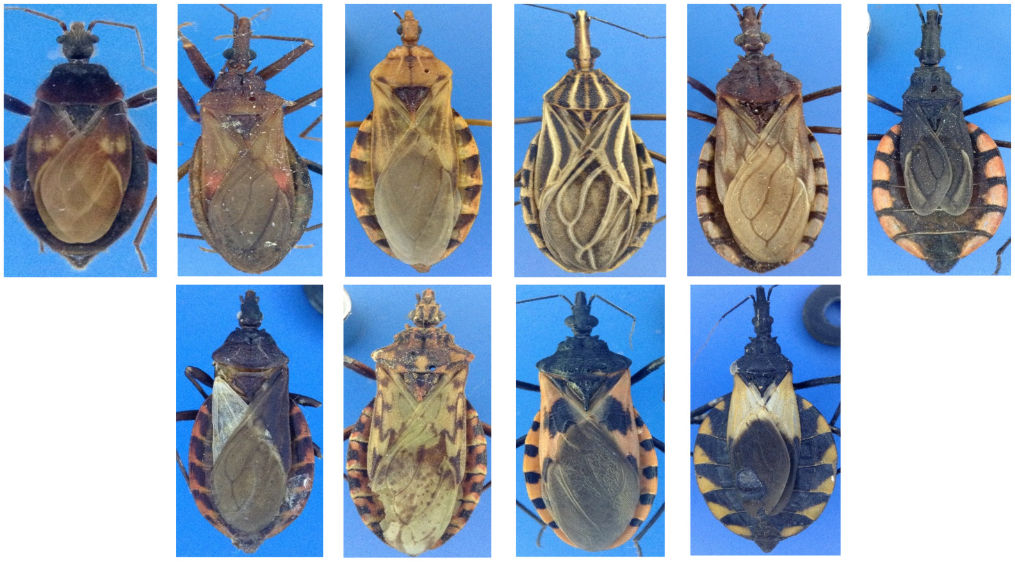


Figure 2.2: First row: Brazilian Triatomine vectors, second row: Mexican Triatomine vectors.

using geometrical features of the insects and neural network algorithm [43] and [44]. Their two papers are discussed in the following sections.

In 2017, Gurgel–Goncalves et al. [43] presented a automatic kissing bugs identification system in 2017. They designed a rather standard but straightforward and precise image capturing device for gathering and selecting high resolution captured kissing bugs at low cost (Figure 2.1 [43]).

The photographs produced using their capturing device are fairly consistent in resolution, orientation, and quality. These qualities enabled full automation of their processing. They captured 1903 images of 67 Brazilian triatomine species and 428 images of 19 Mexican species. But because of shortage in images in some species, they excluded 28 Brazilian and 7 Mexican species from their dataset. Therefor, final proned dataset was categorized as 51 different species from Brazil (39 species) and Mexico (12 species). A total of 1898 images was selected by them, consisting of 1502 Brazilian and 396 Mexican vector images. Example of ten different classes, including six Brazilian kissing bugs and four Mexican kissing bugs, are shown in Figure 2.2.

Their major preprocessing and feature extraction steps include removing (digitally) background and extraneous body parts (legs, antennae), orientation and identification of landmarks, measurements and calculation of ratios, and submission to classification

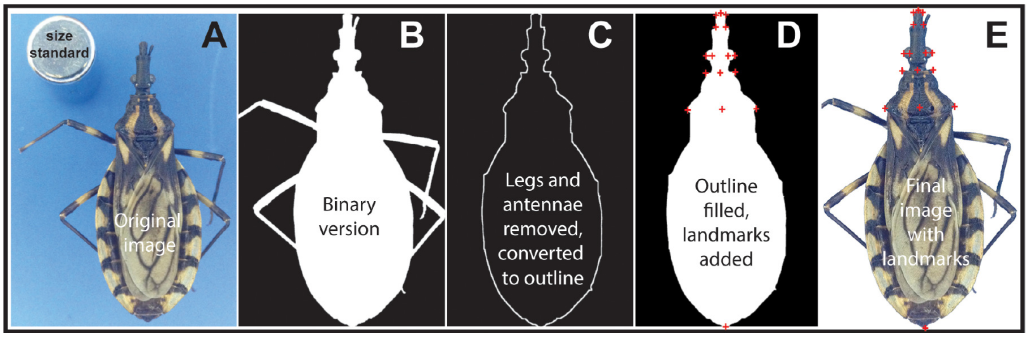

size standard
A
Original image
B
Binary version
C
Legs and antennae removed, converted to outline
D
Outline filled, landmarks added
E
Final image with landmarks

FIGURE 2.3: Preprocessing steps: (A) raw image of a sample vector (B) binary image which is the result of background removal and binarization (C) legs and antennas are removed and edge detection is performed (D) landmarks are extracted from insect's body (E) final image is shown along with extracted landmarks.

phase. For preprocessing, they applied substeps such as: correcting lens distortion, removing background, identifying the edges of specimen's body, clipping the legs and antennas from the image, and smoothing the clipped edges (preprocessing steps are shown in Figure 2.3 [43]).

After that, optimal set of ten geometrical features are extracted. Their extracted features are: total length/clypeus–pronotum forelobe midpoint, total length/mean lateral eye margin, total length/mean eye center, total length/mean lateral pronotum forelobe, total length/pronotum forelobe midpoint to pronotum–humeral angle midpoint, total length/mean lateral pronotum humeral angle, total length/total area, total length/maximum body width, and total length/mean eye center–mean lateral pronotum fore lobe, and a color ratio (the ratio of average gray scale within a defined region of interest (ROI) and outside the ROI to the average grayscale value inside the ROI).

Table 2.1: Summary of species analyzed, sample sizes of photographs, and identification success rates, for the 12 species of Mexican triatomine bugs analyzed in this study.

Species	**Sample Size**	**Success Rate(%)**
Panstrongylus rufotuberculatus (Champion, 1899)	7	100.00
Triatoma barberi Usinger, 1939	29	72.40
Triatoma dimidiata (Latreille, 1811) HG1	44	70.50
Triatoma dimidiata (Latreille, 1811) HG2	30	76.70
Triatoma dimidiata (Latreille, 1811) HG3	40	82.50
Triatoma gerstaeckeri (Stål, 1859)	12	83.30
Triatoma longipennis Usinger, 1939	51	72.50
Triatoma mazzottii Usinger 1941	22	77.30
Triatoma mexicana (Herrich-Schaeffer, 1848)	45	80.00
Triatoma nitida Usinger, 1939	15	46.70
Triatoma pallidipennis Stål, 1872	43	90.70
Triatoma phyllosoma (Burmeister, 1835)	58	46.60

The last feature was helpful in improving the distinguishing power of algorithm for one pair of bugs. For classification, they used a feed–forward neural network using “nn” package in R. Classification is applied separately on Brazilian and Mexican species. A feed–forward neural network was used as a classification algorithm. Their average accuracy was 87.8% and 80.3% for Brazilian and Mexican species respectively. Full nomenclatural details of Mexican and Brazilian species are explained in Tables 2.1 and 2.2 respectively.

TABLE 2.2: Summary of species analyzed, sample sizes of photographs, and identification success rates, for the 39 species of Brazilian triatomine bugs analyzed in this study.

Species	**Sample Size**	**Success Rate (%)**
Cavernicola lenti Barrett & Arias, 1985	15	93.30
Eratyrus mucronatus Stål, 1859	10	80.00
Panstrongylus diasi Pinto & Lent, 1946	30	96.70
Panstrongylus geniculatus (Latreille, 1811)	45	93.30
Panstrongylus lignarius (Walker, 1873)	28	85.70
Panstrongylus lutzi Neiva & Pinto, 1923	34	88.20
Panstrongylus megistus Burmeister, 1835	84	91.70
Psammolestes tertius Lent & Jurberg, 1965	29	100.00
Rhodnius brethesi Matta, 1919	28	96.40
Rhodnius domesticus Neiva & Pinto, 1923	27	96.30
Rhodnius milesi Carcavallo, Rocha, Galvão & Jurberg, 2001	37	89.20
Rhodnius montenegrensis Rosa et al. 2012	39	84.60
Rhodnius nasutus Stål, 1859	73	82.20
Rhodnius neglectus Lent, 1954	60	83.30
Rhodnius pictipes Stål, 1872	43	95.30
Triatoma arthurneivai Lent & Martins, 1940	32	78.10
Triatoma baratai Carcavallo & Jurberg, 2000	29	82.80
Triatoma brasiliensis Neiva, 1911	64	76.60
Triatoma carcavalloi Jurberg, Rocha & Lent, 1998	38	86.80
Triatoma circummaculata (Stål, 1859)	21	85.70
Triatoma costalimai Verano & Galvão, 1958	63	85.70
Triatoma delpontei Romana & Abalos, 1947	29	86.70
Triatoma guazu Lent & Wygodzinsky, 1979	28	64.30
Triatoma infestans (Klug, 1834)	54	83.30
Triatoma juazeirensis Costa & Felix, 2007	21	81.00
Triatoma lenti Sherlock & Serafim, 1967	19	78.90
Triatoma maculata (Erichson, 1848)	39	89.70
Triatoma matogrossensis Leite & Barbosa, 1953	32	75.00
Triatoma melanica Neiva & Lent, 1941	29	79.30
Triatoma pintodiasi Jurberg, Cunha & Rocha, 2013	25	88.00
Triatoma platensis Neiva, 1913	27	74.10
Triatoma pseudomaculata Correa & Espínola, 1964	55	70.90
Triatoma rubrovaria (Blanchard, 1843)	54	59.30
Triatoma sherlocki Papa, Jurberg, Carcavallo, Cerqueira & Barata, 2002	31	93.50
Triatoma sordida (Stål, 1859)	96	81.20
Triatoma tibiamaculata (Pinto, 1926)	41	92.70
Triatoma vandae Carcavallo, Jurberg, Rocha, Galvão, Noireau & Lent, 2002	29	69.00
Triatoma vitticeps (Stål, 1859)	47	85.10
Triatoma williami Galvão, Souza & Lima, 1965	17	70.60

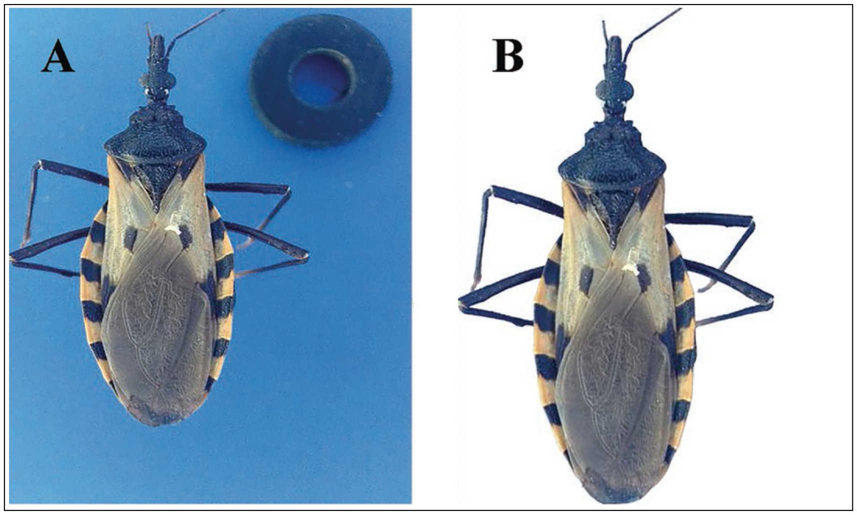


Figure 2.4: An example image of an individual of Triatoma dimidiata HG1. (A) Raw image and (B) final image with background removed digitally.

Later in 2019, the same research team improved their classification results and performance using convolutional neural networks [44, 45]. They used two different sets of raw images and background removed images (cleaned images) as their network's input. With different types of images, they tried to understand how their neural network performs with different image qualities as input (Figure 2.4 [44]). They reported that the accuracy for raw images was almost the same as cleaned images (i.e., 82.9% overall accuracy for raw images, vs 83.0% for cleaned images). Their goal was to be able to compare the results of deep neural network with their previous statistical–based study [43]. Therefore, they decided to use cleaned images as input to have consistency for the sake of result comparison.

They improved their average accuracy in comparison with their previous research to 86.7% and 83.0% for Brazilian and Mexican species respectively. They also applied a paired t–test on outcomes, which produced P–values of 0.028 and 0.025 for Mexican and Brazilian triatomines, respectively. Given that the P–values were less than 0.05, they concluded that the CNN–based results are statistically significantly better than their previous statistical results. For Mexican species, they achieved improved accuracy for 9 out of 12 species and for Brazilian species, their accuracy improved for 23 out of 38 species (Figure 2.5 [44]).

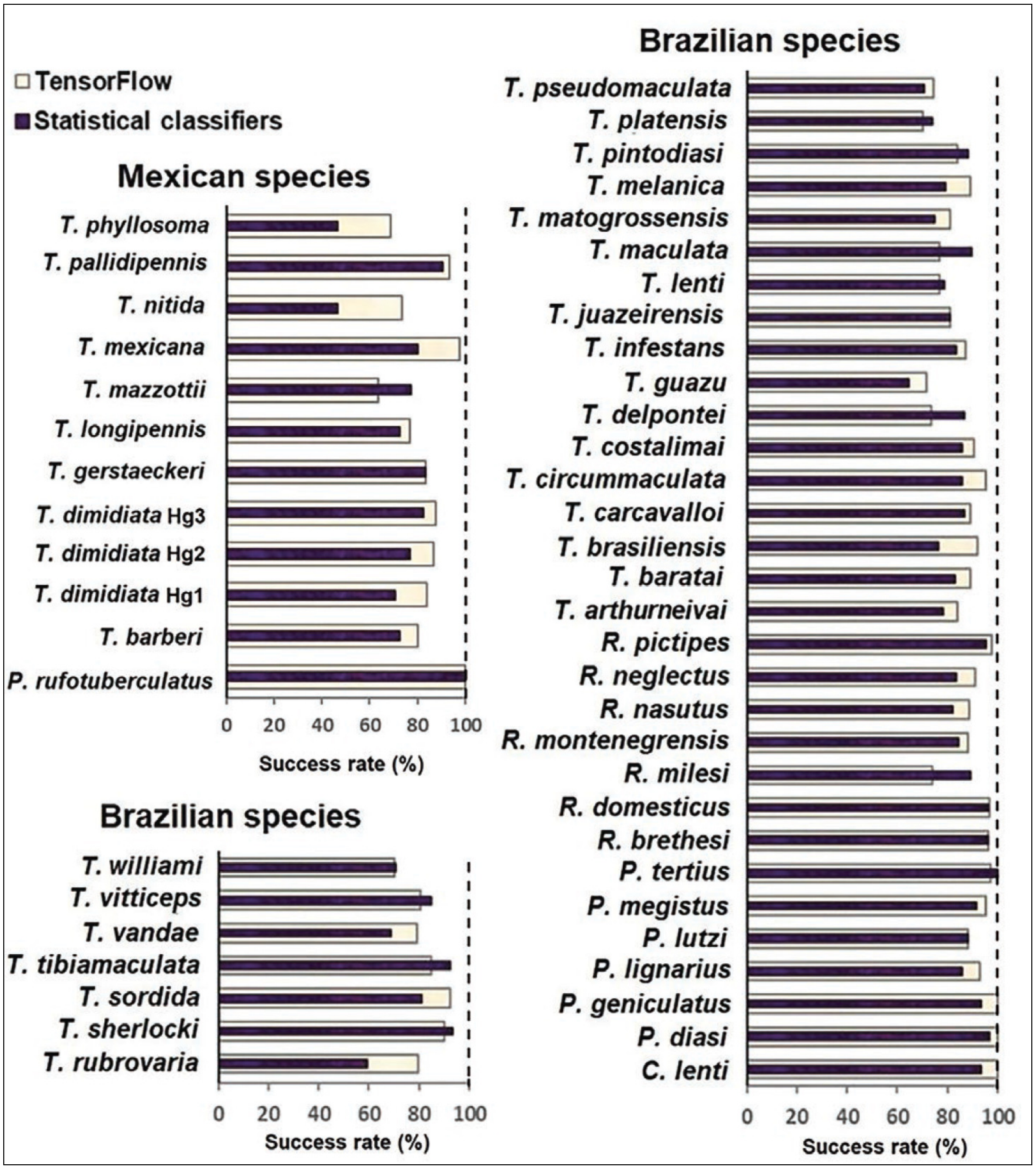


Figure 2.5: Comparison of accuracy rates (%) between deep neural network and statistical classifier at species level for Mexican and Brazilian triatomine vectors.

.

One dilema of the two above mentioned systems is that they are only demonstrated on high–quality images, with high resolution, good lighting, and uniform backgrounds. For example, their dataset only consists of [1936 × 2592] kissing bug images. It is not practical for working with a lower resolution cell phone captured images without perfect lighting and uniformity in the setting. Their second limitation is that their extracted geometrical features need several preprocessing steps using different image filtering and morphological operations. Finally, their third shortcoming is the reported accuracy rates that are 80–88%; which leaves room for improvement.

More recently in 2020, Agany et al. [46] reviewed 32 articles to explore the utilization of data mining techniques towards vector–host–pathogen relationships identification. CD vectors were one of the vector categories they studied and the reference research [44] was one of the reviewed researches. They designed their review using PRISMA workflow [47] which is a design method for literature reviews. PRISMA is utilized to perform a scientific evaluation of the relationships between microbial pathogens, mammalian hosts and arthropod vectors using data mining. Based on their evaluation, even though data mining are being progressively utilized in many different field of life sciences, they do not have taken a root in the field of vector–borne diseases such as CD.

In another research in 2020, Cruz et al. [48] proposed the use of Elliptical Fourier descriptors (EFDs) technique to describe the shape of species and to extract meaningful features from the three species of Triatoma dimidiata (T. dimidiata) which are residing in Mexico and Central America (Colombia, Ecuador, Peru). They utilized the same dataset as [43] and [44] and only considered the images of three classes of their interest in their study. The number of species in the three classes of interest in the reference dataset are 44, 30, and 40 images of the haplogroup 1,2, and 3. They filtered images that were did not have the necessary quality to perform contour analysis and selected 37, 23, and 36 images for the three classes respectively.

They performed their preprocessings manually using Photoshop *CS*5. They removed legs and antennas from each image and only left the body contour. They also changed the brightness and contrast to minimum and maximum values to achieve a binary image. They utilized Shape 1.3 software [49] to evaluate the contour shape based on EFDs. Using this software, they generated the chain code for images. Then for each image, the Fourier transform for 5, 10, 15, 20, 25, and 30 harmonics were computed. They applied this experiments to achieve the best discrimination between the three classes of T. dimidiata species. After that, they applied Principal Component Anlysis (PCA) for dimentionality reduction of features. Then, a discriminative function analysis was performed to select the minimum number of harmonics needed to result in optimal accuracy. Using this techniques, 30 principal components were selected. An ordination discrimination plot was then generated that allowed the best class discrimination. Finally, the confusion matrix was computed to calculated the classification error.

As an alternative technique for classification, a multi–layer perceptron neural network with 30 input and 3 output neurons was designed. The accuracy for 25 harmonics was their best accuracy with 100%, 100%, and 94% accuracy for the three classes respectively. The average accuracy (around 97%) was a significant improvement over the reference paper results with average accuracy of nearly 75% [43] and 86% [44] for the three haplogroups for the first and second reference articles respectively.

Because of the nature of kissing bugs and limitations in gathering many benign and infectious vectors, there is a shortage of sufficient sets of images in this field. he available ones are rather small datasets. As you can see, the only computational trials in this field are using the same dataset to perform their analysis. Due to this shortage, we used the same dataset of images in this research provided by Gurgel et al. [50] consisting of 2030 insect photos; to be able to compare the accuracy rates.

Our methods consist of different steps of preprocessing, feature extraction, feature selection, data balancing and classification stages to achieve higher performance systems. Our methods overcome the first shortcoming by grayscaling and downsizing RGB images to 128 × 128 gray–scale images to perform feature extraction easier and faster. As mentined, the second shortcoming is the numerous preprocessing steps, we resolve it by minimizing the preprocessing steps to only performing background elimination.

We also improved the accuracy by utilizing different feature extraction and classification algorithms. Instead of extracting ten geometrical features, we used the standard PCA algorithm to extract the best 50 and 150 scattered variance–based features in images. We balanced our feature datasets as much as possible using Weka "attributeSelection" and "ClassBalancing" filters. We applied feature selection to optimize the number of features for Mexican and Brazilian datasets separately before applying classification algorithms. Finally, for classification we utilized data mining based algorithms and personalized Decision Tree (DT), Random Forrest (RF), and Support Vector Machine (SVM) for our research. We also applied deep learning based approaches for classification. Deep neural networks have different steps than data mining based classification algorithms. Feature extraction and feature selection steps are omitted for deep neural networks. Our methods are explained in detail in the following chapters.

CHAPTER 3
BACKGROUND CONCEPTS AND TECHNIQUES

As mentioned, our algorithms consist of several steps including preprocessing, feature extraction, feature selection, data balancing, and classification. We evaluate our results using different evaluation metrics. For each step, we utilized different concepts and techniques. In this chapter, these techniques and concepts are explained.

3.1 PREPROCESSING

As mentioned in the previous chapter, one dilemma of previously developed systems is their numerous preprocessing steps. We minimized preprocessing to background removal. Instead of using the same simple thresholding as previously developed systems, which did not have perfectly cleaned results, we used K–means clustering algorithm to remove the backgorund. K–means clustering algorithm is explained in the following subsection. In order to overcome the need for high–guality, high–resolution images, we grayscaled and dowsized the [1936 × 2592] RGB images to [128 × 128] distorted low–quality grayscaled images after background removal by K–means clustering algorithm.

3.1.1 *K–means*

Accurate and efficient background removal is critical for our insect identification research, because the blue background and the white test standard may interfer with the useful body information of kissing bugs. K–means clustering algorithm is the most common algorithm for background removal in many different applicational purposes such as human face and object recognition [51]. K–means is the simplest clustering (unsupervised classification) algorithm that groups data points based on their Euclidean distance from each other. By applying K–means clustering we isolate backgound and test standard object to separate groups from insect's body and remove them from the image. The flowchart of K–means clustering algorithm is shown in

Figure 3.1 [52]. The background and test standard will be zeroed after K–means algorithm ends.

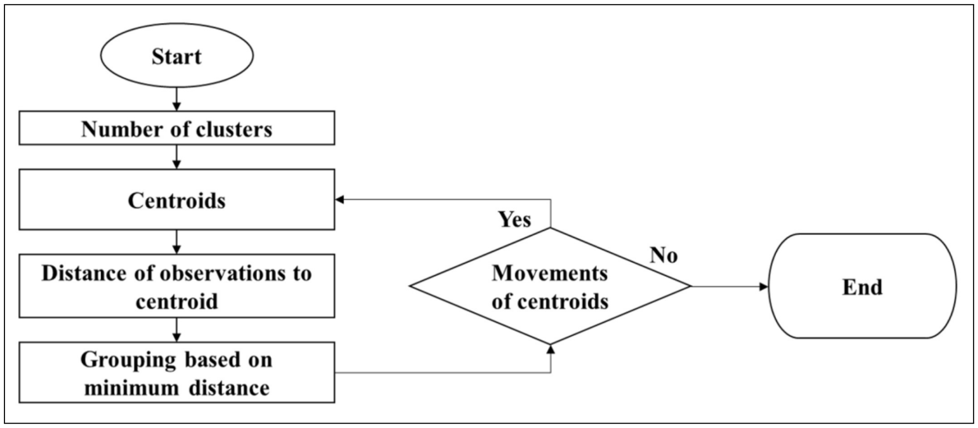

Start
Number of clusters
Centroids
Distance of observations to centroid
Grouping based on minimum distance
Yes
Movements of centroids
No
End

Figure 3.1: K–means clustering flowchart

3.2 FEATURE EXTRACTION

We utilize PCA as our feature selection algorithm. PCA is a standard technique used in statistical pattern recognition and signal processing for data reduction and feature extraction [53]. As the data often contains redundant information, mapping it to a feature vector can eliminate this redundancy and preserve most of the intrinsic information content of the data.

PCA is an efficient technique for extracting an optimal feature vector from high–dimensional data sets [54]. PCA transforms the data set to a new set of ordered orthogonal variables called principal components based on their variances. It extracts the eigenvectors that are associated with the largest eigenvalues from the input distribution. PCA solves the image identification problem within a representation space of lower dimension than image space.

A kissing bug image in 2–dimensions with size $N \times N$ can also be considered as a one dimensional vector of dimension N^2 [55]. For example, original kissing bug image from our database with size 128×128 can be considered a vector of $16,384$, or equivalently a point in a $16,384$ dimensional space. Since the images of kissing bugs have similarities, they are not going to be randomly distributed in this high–

dimentional image space. Therefore, they can be described by a rather low–dimentinal subspace.

The basic idea of the principal components is to find the best orthogonal vectors that represent the distribution of kissing bug images in the whole image space. These vectors define the subspace of images. Each of these vectors is of length N^2, describes an $N \times N$ image, and is a linear combination of the original kissing bug images. In the following subsections, the covariance matrix, eigenvectors and eigenvalues, and PCA are described in detail.

3.2.1 *Covariance Matrix*

In practical pattern recognition problems, there is usually more than one feature available [56]. During the statistical analysis of the data, we have to find out whether these features are independent of one another. Otherwise there exists a relationship between each pair of features. Suppose you extract two features X and Y from a large set of images, to understand whether there exists any relationship between these two features, we have to compute how much the first feature X of each of the patterns in our data set varies from the mean of the second feature Y. This measure, which is computed similar to the variance is called covariance and is always measured between two features. The covariance is computed as follows:

$$Cov(X,Y) = \sum_{i=1}^{n} (X_i - \overline{X})(Y_i - \overline{Y}) \tag{3.1}$$

where n is the number of patterns, and $\overline{X}$ and $\overline{Y}$ are the mean of feature X and Y respectively. If the covariance value is positive, it implies that when one feature (X) increases, the other feature (Y) also increases. If the value of $Cov(X,Y)$ is negative, then as one feature increases, the other one decreases. If there is no correlation between the two features X and Y the covariance becomes zero, indicating that the two features are independent of each other.

In the case of a multi-dimensional feature vector, the covariance is measured between each pair of features. In practical pattern recognition problems, we compute a covariance matrix where each element of the matrix gives a measure of the covariance between two features.

3.2.2 *Eigenvectors and Eigenvalues*

Before we discuss principal component analysis we will briefly explain the concept of eigenvectors and eigenvalues of a matrix [56]. Let us assume that we have a square matrix A of dimension $n \times n$, which when multiplied by a vector X of dimension $n \times 1$ yields another vector Y of dimension $n \times 1$, which is essentially the same as the original vector X that was chosen initially. Such a vector X is called an eigenvector which transforms a square matrix A into a vector, which is either the same vector X or a multiple of X (i.e., a scaled version of the vector X). The matrix A is called a transformation matrix, while the vector X is called an eigenvector. As is well known, any integer multiplication of the vector results in the same vector pointing to the same direction, with only its magnitude being scaled up (i.e., the vector is only elongated).

Eigenvectors can be determined only from the square matrices, while every square matrix does not necessarily yield an eigenvector. Also, an $n \times n$ square transformation matrix may have only n number of eigenvectors. All these eigenvectors are orthogonal to each other. Every eigenvector is associated with a corresponding eigenvalue. The concept of an eigenvalue is that of a scale which when multiplied by the eigenvector yields the same scaled vector in the same direction.

3.2.3 *Principal Component Analysis*

While computing the principal component analysis [56] we represent an $N \times N$ image as a one-dimensional vector of N^2 elements, by placing the rows of the image one after another, and then stacking the acquired vectors in rows on top of each other in a dataset. Then we compute the covariance matrix of the entire data set. Next, we compute the eigenvalues of this covariance matrix. The eigenvectors corresponding to the most significant eigenvalues will yield the principal components. To get the original data back we have to consider all the eigenvectors in our transformation. If we discard some of the less significant eigenvectors in the final transformation, then the retrieved data will lose some information. However, if we choose all the eigenvectors we can retrieve the original data.

A simple example of PCA for a multivariate gaussian distribution is shown in Figure 3.2. The vectors shown are the eigenvectors of the covariance matrix scaled by the square root of the corresponding eigenvalue and shifted, so their tails are at the mean.

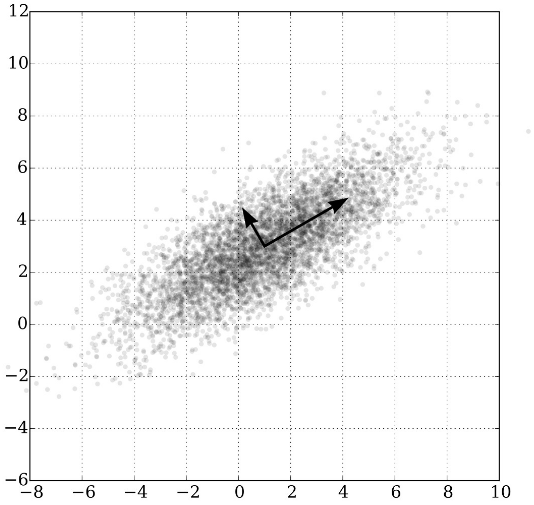


Figure 3.2: PCA of a multivariate Gaussian distribution centered at (1,3) with a standard deviation of 3 in roughly the (0.866, 0.5) direction and of 1 in the orthogonal direction.

3.3 FEATURE SELECTION

Feature selection is applied on PCA feature set. Although PCA is an effective algorithm for finding orthogonal features, extracting too many eigenvectors might result in extracting redundant features. Therefore, a feature selection algorithm will be applied to optimize the feature set.

There are different arrtibute selection filters in Weka which examine attributes based on different criterias such as Correlation based Feature Selection, GainRatio Attribute evaluation and InformationGain Attribute evaluation.

Correlation–based attribute selection methods evaluate the attributes with respect to the target class. Pearson's correlation techniques is utilized to measure the correlation between each attribute and the target class. GainRatio Attribute evaluation method measures the significance of attributes with respect to target class on the basis of gain ratio. Gain Ratio is computed by the following equation where H represents the Entropy.

$$GainR(Class, Attribute) = \frac{(H(Class) - H(Class|Attribute))}{H(Attribute)} \tag{3.2}$$

Finally, Information Gain Attribute evaluation method measures the significance of attribute by the information gain factor calculated with respect to target class. Information gain is computed by the following equation where H represents the Entropy.

$$InfoGain(Class, Attribute) = H(Class) - H(Class|Attribute) \tag{3.3}$$

CfsSubsetEvaluator is an attribute (feature) selection filter. It evaluates the value of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. The highly correlated subsets of feature with the class with low intercorrelation are preferred and selected. BestFirst algorithm uses a greedy hillclimbing with backtracking search algorithm to explore different subset of features in the space. BestFirst module is consists of two submodules: attribute evaluator and search method. Each submodule has multiple techniques as options. The attribute evaluator is the technique by which each attribute in the dataset is evaluated in the context of the output variable (e.g., the class). The search method is the technique by which to try or navigate different combinations of attributes in the dataset to arrive at a short list of chosen features.

3.4 DATA BALANCING

Traditional data mining methods assume a balanced distribution of classes. Several real data sets suffer from class–imbalance problem where it has a rare positive example but many negative ones. For example, when learning from the results of a medical test the vast majority of instances have a negative outcome and only a few return positive. Some machine learning algorithms will learn to ignore the minority class and classify all cases into the majority class because this will trivially yield high classification accuracy. In other words, many classification algorithms are vunarable to unbalanced data and their performance severaly drops in presenece of imbalanced dataset. There are different data balancing techniques that aim at addressing this data problem. the two most common techniques for overcoming this problem are undersampling of the majority class and oversampling of the minority class [57].

3.4.1 *Undersampling the majority class*

In the case of having large datasets with a lot of data points, getting rid of some data points in majority classes not only does not cause any harm for classification process, but also helps in increasing the performance of the classification algorithm by balancing the data in different classes. Two instance–based Weka filters can be used to implement the undersampling of the majority class called "Resample" and "SpreadSubsample" filters. The following expression is utilized in the Resample module for determining the number of instances to sample for particular class i:

$$\begin{aligned} sampleSize = \frac{m\_SampleSizePercent}{100.0} \times \\ ((1 - m\_BiasToUniformClass) \times numInstancesPerClass[i] + \\ m\_BiasToUniformClass \times \frac{numOfInstances}{numActualClasses})) \end{aligned} \tag{3.4}$$

Where *numOfInstances* gives the total number of instances in the dataset, *numInstancesPerClass[i]* holds the number of instances in class i and *numActualClasses* corresponds to the number of classes that occur in the dataset (some declared classes in an ARFF file might not have any samples in the data). Therefore, to undersample the majority class in the way that both classes have the same number of instances, the configuration values of the filter are *biasToUniformClass*=1.0 and *sampleSizePercent=X*, where $X/2$ is nearly the percentage of data that belongs to the minority class. We also need to configure the filter to perform sampling without replacement by applying *noReplacement=true*. SpreadSubsample is a much easier way to achieve the same effect. Only the 'M' parameter, which is a distribution spread, needs to be set to 1.0.

3.4.2 *Oversampling the minority class*

In the case of having samll datasets with few data points, undersampling is harmful for the overall classification process. Therefore, instead of getting rid of data points, data regenarating techniques (oversampling techniques) should be considered to balance the data through increasing the dataset size. In order to oversample the minority class in the way that both classes have the same number of instances, the configuration values of the Weka Resample filter are *noReplacement=false*, *biasToUniformClass*=1.0, and *sampleSizePercent*=Y, where $Y/2$ is nearly the percentage of data that belongs to the majority class. We utilize this method as a preprocessing filter before classification phase to increase the overall number of instances in our insect image dataset and balance the number of images per class [58].

3.5 CLASSIFICATION

We apply different data mining based and deep learning based algorithms for classification. These algorithms are; Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), and deep neural networks. These algorithms are explained in following subsections.

3.5.1 *Decision Tree*

A DT is a decision support system that uses tree-like graph decisions and possible after–effects, including chance event results, resource costs, and utility. A Decision Tree, or a classification tree, is used to learn a classification function that concludes the value of a dependent attribute (variable) given the values of the independent (input) attributes (variables) [59]. The DT algorithms classify the examples by sorting them down the tree starting from the root node to some leaf/terminal node. The leaf/terminal node is providing the classification of the example. The process is recursive and will repeat for every subtree rooted at the new node. You can see the simple description ofthe basic concept of the decision tree in Figure 3.3.

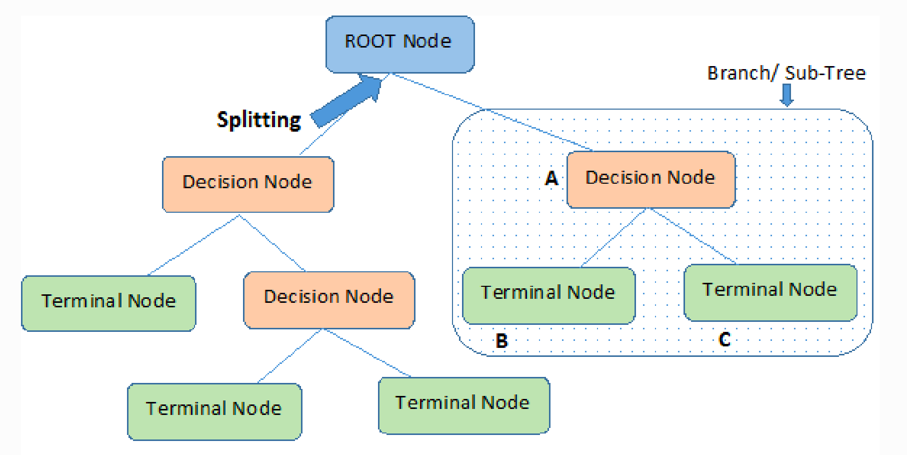


FIGURE 3.3: Basic DT concepts. Note: Node A is parent of Nodes B and C.

Decision trees are very reliable approaches in knowledge discovery and data mining. They are capable of processing large and complex masses of data and discover useful patterns. Decision trees are highly effective tools in many areas such as data and text mining, information extraction, machine learning, pattern recognition, and also bug recognition [60].

3.5.2 *Random Forest*

The RF classifier is an ensemble classification method that works by generating several decision trees from bootstrap samples of the training data. Each tree votes for the most popular class to classify an input vector [61]. It is a combination of tree predictors in which decision trees are constructed using resampling technique with replacement, the inducers randomly samples the attributes and chooses the best split among those variables rather than the best split among all attributes [62]. The main steps of Random Forest are described in Table 3.1.

Table 3.1: Algorithmic steps of Random Forest.

```
Inputs: DTI (a decision tree inducer), T (the iterations numbers), S (train sets),
        r (sampling ration), N (number of attributes used in each tree)
Train: for i = 1 to T
        Get sample St from S with replacement using r;
        Build classifier Mt based on the inducer randomly samples N
        of the attributes and choose the best split
Classification: : new instance classified by classifiers $M_{t(t=1,...,T)}$
        then performed using majority vote.
```

The assignment of class label of an unknown instance is performed using majority voting strategy. Each tree votes for the most popular class to classify an input vector. The output of the classifier is chosen by taking the majority voted class from all the trees in the forest. The process of majority voting is shown in Figure 3.4 [63].

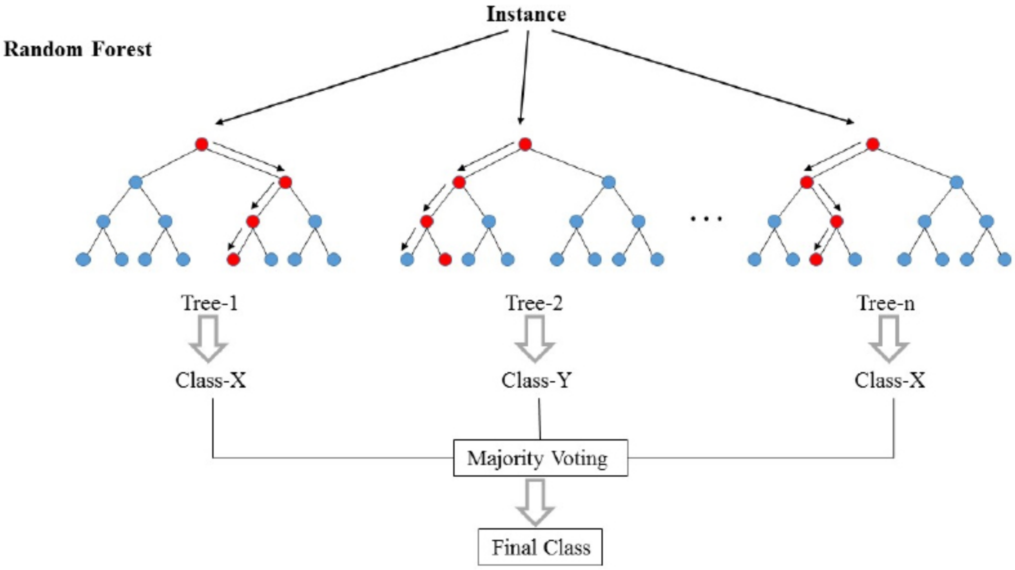


Figure 3.4: Example of majority voting on the final class in RF.

RF is computationally less intensive than other ensemble approaches. It is robust to noise and outliers and can handle many input variables without overfitting. Due to the important advantages such as handling very large number of input attributes and low time cost, RF is widely used in image classification research.

3.5.3 *Support Vector Machine*

Although SVM was proposed by Vapnik in the late 1960s, it has not received significant attention until recent years when it has become a promising estimator in data–driven fields. SVM is a supervised method to perform dichotomy classification of multidimensional feature–vectors [64]. SVM is a discriminative classifier that generates a hyperplane that separates the data points into two classes with maximal margin [65]. The margin is defined as the distance between hyperplane and nearest points. The best hyperplane with largest margin is usually selected.

The original algorithm was developed as a linear classifier (See Figure 3.5 [65]) which was not efficient to provide a good discrimination between classes because many datasets demonstrate complex structure. Therefore, it was generalised to a non–linear classifier using several non–linear kernel functions (e.g. polynomial, hyperbolic tangent (sigmoid) and radial basis function (RBF)).

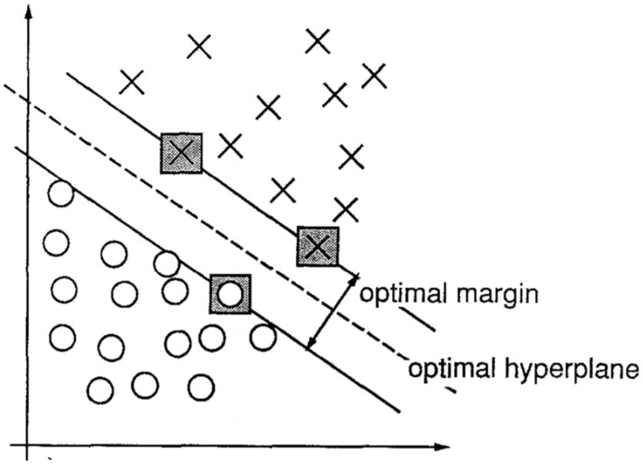


FIGURE 3.5: An example of a separable problem in a 2 dimensional space. The support vectors which are marked with grey squares, define the the largest margin of separation between the two classes.

The basic idea of SVM method is to transform the input features into a higher–dimensional space so that the two classes are linearly separated by a high–dimensional surface, known as hyper–plane [64]. Given a training dataset $\{x_n\}_{n=1}^{N}$ with N samples, where $x \in \mathbb{R}^L$ is a vector of L input–features, and its corresponding known output–features $\{y_n\}_{n=1}^{N}$, with $y_n \in \{-1, 1\}$, the SVM model is defined then as:

$$f(x) = \mathbf{w}^T \phi(\mathbf{x}) + b \tag{3.5}$$

where $\phi : x \to \phi(x) \in \mathbb{R}^L$ is any non–linear function that maps the input data into the high–dimensional feature space with $H \geq L$. Originally, assuming linearly separable features, this function was trivially defined as $\phi(x) = x$. The unknown parameters of the model are w, a weight vector which is normal to the hyper–plane and b, the hyper–plane bias.

The SVM model is defined then to cope with nonseparable features by allowing misclassification errors. Therefore, the SVM model presented above is subject to the following constrains:

$$\begin{aligned} y_n - f(\mathbf{x}_n) &\leq \zeta_n + \varepsilon \\ f(\mathbf{x}_n) - y_n &\leq \zeta_n^* + \varepsilon \\ \varepsilon, \zeta_n, \zeta_n^* &\geq 0, \forall n \end{aligned} \tag{3.6}$$

where ε is the (in) sensitivity, i.e. the maximum misclassification error allowed and $\{\zeta_n, \zeta_n^*\}_{n=1}^{N}$ are slack variables quantifying the output–features deviation from the positive and negative classes.

The optimisation of the previous model, subject to the soft–margin constrain, defines a hyper–plane which separates the training data with the maximum margin. The optimisation problem can be solved by using the Lagrange multipliers method, (for details see [66]), yielding to the next cost function:

$$L(\{a_n, a_n^*\}_{n=1}^{N}) = -\frac{1}{2} \sum_{i,j=1}^{N} (a_i - a_i^*)(a_j - a_j^*) K(\mathbf{x}_i, \mathbf{x}_j) - \varepsilon \sum_{i=1}^{N} (a_i + a_i^*) + \sum_{i=1}^{N} (a_i + a_i^*) y_i \tag{3.7}$$

where $\{a_n, a_n^*\}_{n=1}^N$ are the Lagrange multipliers and $K(x_i, x_j)$ is the Kernel function, defined as the inner product of the transformed input–feature vectors:

$$K(\mathbf{x}_i, \mathbf{x}_j) :=< \phi(\mathbf{x}_i)|\phi(\mathbf{x}_j) > \tag{3.8}$$

The optimisation of this cost function is significantly simplified by introducing the kernel notation. Instead of designing a mapping function, then transform the data and later compute the inner products, the SVM approach directly defines the kernel as a function of the input–feature vector. Some kernel functions typically considered on SVM applications are shown below:

$$\begin{aligned} K_{linear}(x, x^{'}) &= x.x^{'} \\ K_{polynomial}(x, x^{'}) &= (\gamma x x^{'} + r)^{\rho} \\ K_{RBF}(x, x^{'}) &= exp(-\gamma||x - x^{'}||^2) \\ K_{sigmoid}(x, x^{'}) &= tanh((\gamma x x^{'} + r)) \end{aligned} \tag{3.9}$$

Once we estimate $\{\hat{a}_n, \hat{a}_n^*\}_{i=1}^N$ by maximising the cost function defined above, the margin can be inferred as:

$$\hat{\mathbf{W}} = \sum_{n=1}^{N} (\hat{a}_n - \hat{a}_n^*)\phi(\mathbf{x}_n) \tag{3.10}$$

such as f(x) can be directly estimated as:

$$\hat{f}(x) = \sum_{n=1}^{N} (\hat{a}_n - \hat{a}_n^*)K(\mathbf{x}_i, \mathbf{x}) + \hat{b} \tag{3.11}$$

where the computation of $\hat{b}$ can be conveniently dropped out by preprocessing and centralising the data, forcing the bias to be zero. An example of non–linear SVM classification with kernel is shown in Figure 3.6 [67].

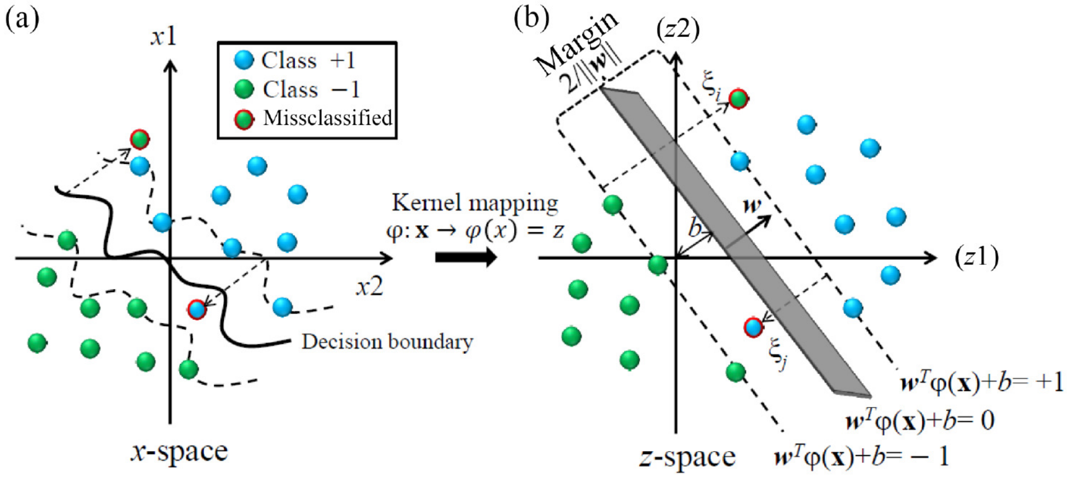


Figure 3.6: Graphical representation of the SVM classifier with a non–linear kernel, (a) complex binary pattern classification problem in input space, and (b) non–linear mapping into high–dimensional feature space where a linearly separable data classification takes place.

3.5.4 *Convolutional Neural Networks*

Artificial neural networks (ANNs) are the key building blocks for modern computer vision systems [68]. ANNs consist of a collection of "neurons" (i.e., nodes) and edges that connect these neurons. If there is a connection between two neurons, then the first neuron's output serves as input for the second neuron. Every connection has an associated weight that signifies the relative importance of the input. A neuron performs a computation on the weighted sum of its inputs. This computation is known as an activation function—for instance, a commonly used activation function is the Rectified Linear Unit (ReLU), which applies the transformation $f(x) = max(0, x)$ (equivalent to replacing negative values with 0). The output of the neuron is then passed along to the other neurons to which it is connected. The neural networks used in computer vision are generally feed forward networks, whereby neurons are arranged in layers, and all connections flow in a single (forward) direction. In other words, neurons in the same layer have no connections with one another. Instead, they only have connections with neurons in adjacent layers, receiving inputs from the preceding layer, and sending outputs to the following layer. The most commonly used type of feed-forward ANN is the multilayer perceptron, also known as a fully connected layer. As its name suggests, every neuron in a fully connected layer has connections to every neuron in the preceding layer.

The currently best–performing algorithms for feature extraction and image classification use convolutional neural networks (CNNs) [69, 70]). CNNs build upon ANNs by including layers that perform convolution operations, which serve to extract features from input images. Any image can be represented as a matrix of pixel values. Convolution operations use these pixel values to calculate new values using element–wise matrix multiplication with a small matrix (aka a "filter" or "kernel") that sweeps over original image pixel values (Figure 3.7 [68]). Then, the sums of the element–wise multiplications form the elements of a new matrix of convolved features (also known as an "activation map" or "feature map").

As convolution operations are linear, a ReLU layer is usually applied following convolution in order to introduce non–linearity into the network. This step is important because a simple linear function is limited in its ability in capturing complex mappings between the input (images) and output (classes). Although other nonlinear activation functions exist, ReLU has been shown to perform better in most situations [71]. Following the convolution and ReLU layers are pooling layers that are used to perform downsampling (i.e., dimension reduction), removing extraneous features while retaining the most relevant information. Commonly used pooling operations include max pooling (whereby the highest value in a neighborhood of pixels is retained, and all others discarded) and average pooling (whereby the average of all values in a neighborhood of pixels is calculated and retained).

An image can be represented as a matrix of pixel (px) values. At part(a) of the figure we have a 4px by 4px image represented as a 4×4 matrix. We use an example filter, or kernel, that is represented by the 2×2 matrix shown. In part(b) Convolution is performed by sweeping the filter across the image and summing the resulting values from element–wise multiplication of the values of the image matrix that the filter overlaps with the corresponding filter values. Then, the sum values are saved to a new matrix that has one entry for every step of the convolution process. The stride value of 1px is used in this process. It means that the filter moves 1px in each step. This is repeated until the filter has been passed over the entire image. In part(c) the

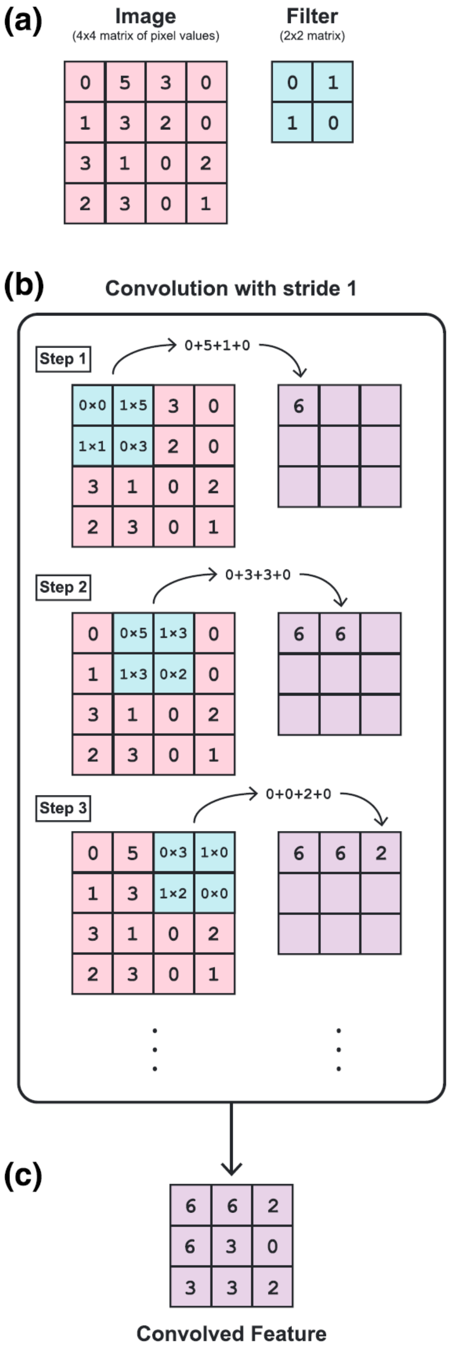


Figure 3.7: Example of how convolution is performed in a convolutional neural network.

resulting matrix of sums is the convolved feature, also known as an "activation map" or "feature map".

Combining, the convolution, ReLU, and pooling layers comprise the feature extraction portion of the CNN, producing the high–level features that are then used to perform classification. The values computed by the network are then processed using fully connected layers, generating a vector of probabilities reflecting the probability that a given image falls in any given class. This complete process (from input to feature extraction to classification) is known as forward propagation [68].

The training set provides CNN with known examples of the correct mapping between image values and weights and the final classification (i.e., the true probability vector). When CNN is initialized, all weights and filters are randomly assigned. The network then takes the input images and runs the first forward propagation step. As the weights and filters are random at this point, the output is a vector of random class probabilities for each image. The total error (i.e., the sum of the differences between the true probability vector and the output probability vector) is calculated. The network then performs back propagation, which is the process of updating all the weights and filters using gradient descent to minimize the total error. One complete forward propagation and back propagation of the entire data set is called an epoch.

Ideally, all images would be passed through the neural network at once to result in the most accurate back propagation updates. However, in practice this is computationally intractable, and the data must be broken up into separate smaller batches to feed into the network. In general, the larger the batch size the better training process. However, the maximum batch size is limited by the amount of memory available to hold all of the data at once. The number of batches required to complete a single epoch is called the number of iterations. For example, a data set containing 1,000 images could be split into five batches of 200 images. Training a CNN using this data set would then take five iterations to complete one epoch. The number of epochs required to train a network adequately is variable and depends on the characteristics of the data set and the parameters associated with the gradient descent algorithm being used [68].

By updating weights and kernels in the back propagation to reduce classification error, the network learns how to classify the training images accurately, building an association between a particular collection of weights and kernels and a particular output class. The best performing model is then used to classify the images in the validation data set. The performance of the model on the validation set gives us an idea of how well the model performs and what sort of accuracies we might expect if the model was used to classify entirely novel images. Model performance is evaluated by looking at validation accuracy (i.e., the proportion of images in the validation set that is correctly identified by the trained model) and validation loss (i.e., the sum of errors for each image in the validation set, where the error is determined by a loss function such as cross-entropy).

3.5.5 *VGG16*

The 16–layer VGG16 (named after the Visual Geometry Group at Oxford University) CNN [72] is a commonly used image classification neural network. Although VGG16 is a relatively shallow network, its development was critical in showing that, in general, the deeper a neural network (i.e., the more layers it contains), the more accurate its performance. However, training difficulty and computational costs (i.e., time) increase with neural network depth. The structure and architecture map of VGG16 are shown in Figures 3.8 and 3.9 respectively.

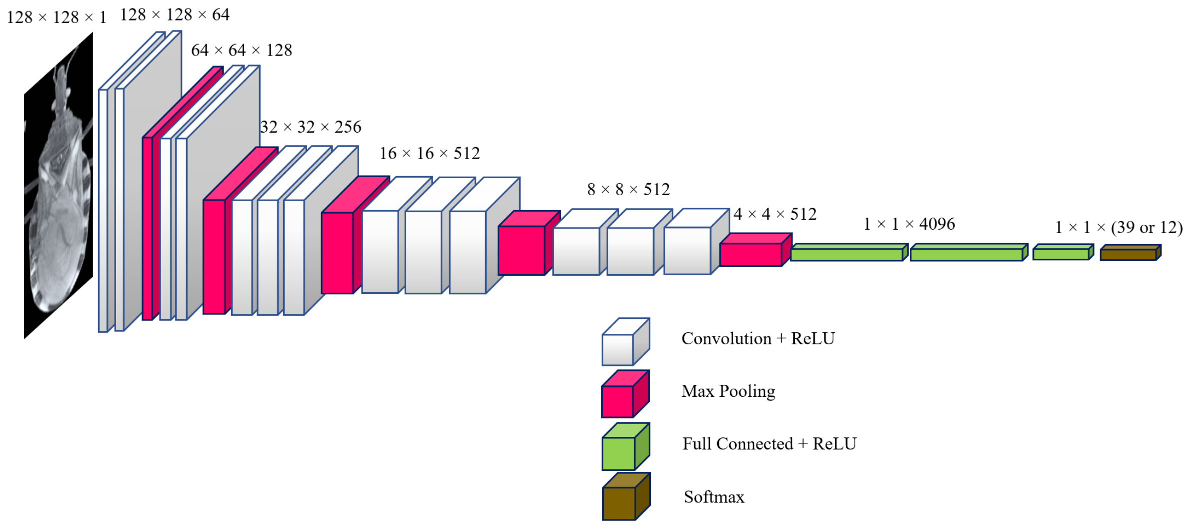


Figure 3.8: VGG-Net-D (VGG16) – convolutional neural network for image classification

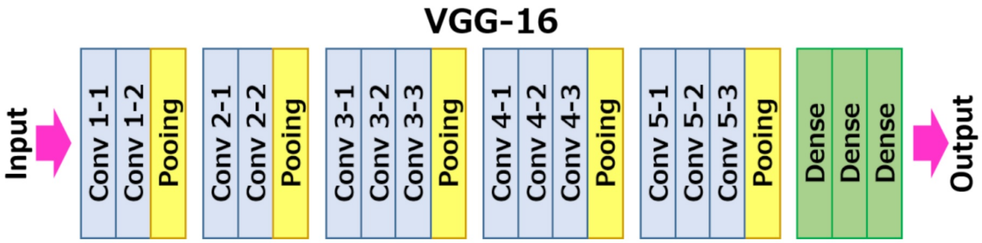


Figure 3.9: VGG-Net-D (VGG16) – Architecture map

3.5.6 *ImageNet checkpoint*

ImageNet is a dataset of over 15 million labeled high–resolution images belonging to roughly 22, 000 categories. The images were collected from the web and labeled by human labelers using Amazon's Mechanical Turk crowd-sourcing tool. Starting in 2010, as part of the Pascal Visual Object Challenge, an annual competition called the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) has been held. ILSVRC uses a subset of ImageNet with roughly 1000 images in each of 1000 categories. There are approximately 1.2 million training images, 50, 000 images for validation, and 150, 000 images for testing. ImageNet is widely used for deep neural networks via the concept of transfer lerning explained in the next subsection. In Figure 3.10 some class samples of ImageNet dataset are shown.

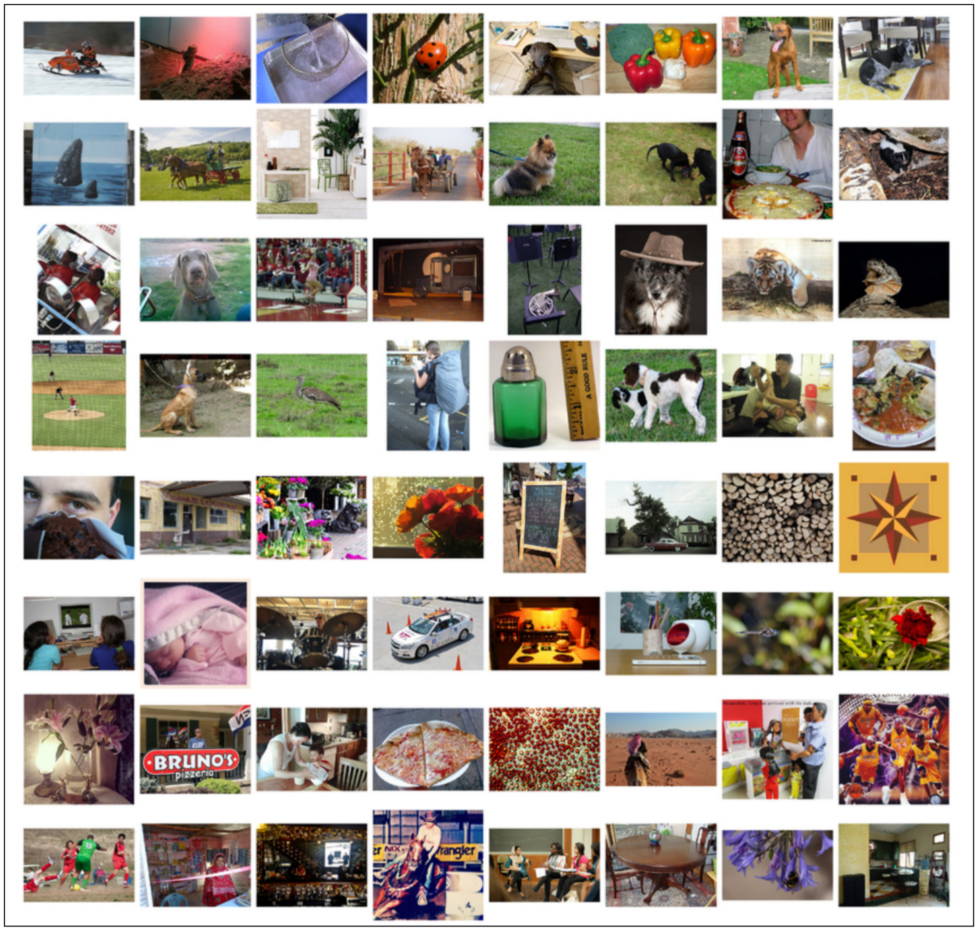


FIGURE 3.10: Some sample classes of ImageNet dataset

3.5.7 *Transfer Learning*

One technique that eases the computational burden and allows for robust models to be trained using relatively small data sets is transfer learning. Transfer learning uses weights from a model previously trained using another data set on a new task; these weights are "frozen" in the new model so that they are not trainable, thus reducing the number of parameters that must be estimated [68]. New images are then used only to train the unfrozen layers at the end of the CNN to fine–tune the model to the task at hand. This approach can be an efficient and effective strategy when one does not have a huge data set to train a CNN from scratch. For example, ImageNet has been used to train many CNNs, and the resulting weights are freely available. Transfer learning allows accurate models to be trained with hundreds to thousands, rather than millions of images.

3.6 EVALUATION METRICS

To evaluate the performance of our classifiers, we used several known common metrics. These metrics are explained in this section. There are four basic terms we need to know that are used in computing many evaluation measures [73].

- **True Positives** (TP): These refer to the positive samples that were correctly labeled by the classifier as possitive.
- **True Negatives** (TN): These refer to the negative samples that were correctly labeled by the classifier as negative.
- **False Positives** (FP): These refer to the negative samples that were incorrectly labeled by the classifier as possitive.
- **False Negatives** (FN): These refer to the positive samples that were incorrectly labeled by the classifier as negative.

These terms are summerized in the confusion matrix of Figure 3.11 [73].

		Predicted class		
		yes	*no*	Total
Actual class	*yes*	*TP*	*FN*	P
	no	*FP*	*TN*	N
	Total	P'	N'	$P+N$

FIGURE 3.11: Confusion matrix, shown with totals for positive and negative samples.

A confusion matrix is a useful tool for analyzing how well a classifier can recognize samples of different classes. Values *TP* and *TN* tell us when the classifier is getting things right, while *FP* and *FN* tell us when the classifier is getting things wrong (i.e., mislabeling). Given m classes (where $m \geq 2$), a confusion matrix is a table of at least size m by m. An entry, $CM_{i,j}$ in the first m rows and m columns indicate the number of samples of class i that were labeled by the classifier as class j. For a good classifier with high accuracy, most of the samples would be represented along the diagonal of the confusion matrix, from the entry $CM_{1,1}$ to entry $CM_{m,m}$, while the rest of the entries are zero or close to zero. Therefore, FP and FN will be around zero. The table might have additional rows or columns to provide total values. For example, in the confusion matrix of Figure 3.11, P and N are shown. P is the number of positive samples and N is the number of negative samples. Also, $P^{'}$ is the number of samples that were labeled as positive $(TP + FP)$ and $N^{'}$ is the number of samples that were labeled as negative $(TN + FN)$. The total number of samples is $TP + TN + FP + TN$, or $P + N$, or $P^{'} + N^{'}$. Note that although the confusion matrix shown is for a binary classification problem, confusion matrices can be easily similarly drawn for multiple classes. The evaluation metrics used in our research are accuracy–based measures such as precision, recall, f–measure, and accuracy. These measures are meaningful combinations of confusion matrix basic terms.

3.6.1 *Precision*

Precision can be thought of as a measure of exactness (i.e., what percentage of samples labeled as positive are actually such). It is defined as Equation 3.15.

$$precision = \frac{TP}{TP + FP} \tag{3.12}$$

A perfect precision score of 1.0 for a class C means that every sample that the classifier labeled as belonging to class C does indeed belong to class C. However, it does not tell us anything about the number of class C samples that the classifier mislabeled.

3.6.2 *Recall*

Recall is a measure of completeness (what percentage of positive samples are labeled as such). It is defined as Equation 3.13.

$$recall = \frac{TP}{TP + FN} = \frac{TP}{P} \tag{3.13}$$

A recall is the same metric as sensitivity or true positive rate (TPR). A perfect recall score of 1.0 for class C means that every item from class C was labeled as such, but it does not tell us how many other samples were incorrectly labeled as belonging to class C. There tends to be an inverse relationship between precision and recall, where it is possible to increase one at the cost of reducing the other. Therefore, precision and recall scores are typically used and reported together.

3.6.3 *F–measure*

An alternative way of utilizing precision and recall is to integrate them into a single measurement. This approach is called the F measure (also known as the F1 score or F–score). It is defined as Equation 3.14.

$$F = \frac{2 \times precision \times recall}{precision + recall} \tag{3.14}$$

The f–measure is the harmonic mean (average) of the precision and recall factors. It gives an equal weight to precision and recall values.

3.6.4 *Accuracy*

On a given test set, the accuracy of a classifier is the percentage of test set samples that are correctly classified by the classifier. That is,

$$accuracy = \frac{TP + TN}{P + N} \tag{3.15}$$

3.7 CROSS VALIDATION

In k–fold cross–validation the initial data are randomly divided to k mutually exclusive subsets, or "folds", $D_1, D_2, ..., D_k, i = 1...k$ each of nearly equal size. Training and testing are performed k times. In iteration i, partition D_i is reserved as the test set, and the remaining partitions are collectively used to train the model. That is, in the first iteration, subsets $D_2, ..., D_k, i = 2...k$ collectively serves as the training set to obtain a first model, which is tested on D_1; the second iteration is trained on subsets $D_1, D_3, ..., D_k, i = 1, 3...k$ and tested on D2; and so on. In k–fold cross–validation, each sample is used the same number of times for training and once for testing. For classification, the accuracy measurement is the overall number of correctly classified samples from the k iterations, divided by the total number of samples in the initial data.

Leave–one–out is a special case of k–fold cross–validation where k is set to the number of initial samples. It means that only one sample is "left out" at a time for the test set. In cases that the number of samples is few and we need to get the most out of our dataset to train the classifier, this method is used. In this work, we used 10–fold cross–validation for our evaluations to get unbiased results. It means in 10 different iterations, 90% of images are used in training, and 10% of samples are used in testing the classifier.

3.8 MODEL COMPARISON USING STATISTICAL TESTS OF SIGNIFICANCE

Suppose that two classification models are being generated from the data and are named M_1 and M_2. The 10–fold cross–validation is performed to obtain a mean error rate for each. It is possible to have considerable variance between error rates within any given 10–fold cross–validation experiment. Although the mean error rates obtained for M_1 and M_2 might seem different, the difference may not be statistically significant. To determine if there is any "real" difference in the mean error rates of two models, we need to employ statistical significance. Also, we want to obtain some confidence limits for our mean error rates.

For a given model, the individual error rates calculated in each round of 10–fold cross–validation may be considered different, independent samples from a probability distribution. In general, they follow a t–distribution with $k-1$ degrees of freedom where, here, $k = 10$. (This distribution looks very similar to a normal, or Gaussian, distribution even though the functions defining the two are quite different. Both are unimodal, symmetric, and bell–shaped). This approach allows us to do hypothesis testing where the significance test used is the t–test or Student's t–test. We hypothesize that the two models are the same, or in other words, that the difference in the mean error rate between the two is zero. If we can reject this hypothesis (referred to as the null hypothesis) we can conclude that the difference between the two models is statistically significant and we can select the model with the lower error rate.

In data mining practice, we may often employ a single test set; that is, the same test set can be used for M_1 and M_2. In such cases we make a pairwise comparison of the two models for each 10–fold cross–validation round. That is, for the i^{th} round of 10–fold cross–validation, the same cross–validation partitioning is used to obtain an error rate for M_1 and M_2. Let $err(M_1)_i$ (or $err(M_2)_i$) be the error rate of model M_1 (or M_2) on round i. The error rates for M_1 are averaged to obtain a mean error rate for M_1, denoted $\overline{err}(M_1)$. Similarly, we can obtain $\overline{err}(M_2)$. The variance of the difference between the two models is denoted $var(M_1 - M_2)$. The t–test computes the t–statistic with $k - 1$ degrees of freedom for k samples. In our example, we have $k = 10$ since, here, the k samples are our error rates obtained from ten 10–fold cross–validations for each model. The t–statistic for pairwise (paired t–test) comparison is computed as follows:

$$t = \frac{\overline{err}(M_1) - \overline{err}(M_2)}{\sqrt{\frac{var(M_1 - M_2)}{k}}} \tag{3.16}$$

where

$$var(M_1 - M_2) = \frac{1}{k}\sum_{i=1}^{k}[err(M_1)_i - err(M_2)_i - (\overline{err}(M_1) - \overline{err}(M_2))]^2 \tag{3.17}$$

To determine whether M_1 and M_2 are significantly different, we compute t and select a significance level (sig). In practice, a significance level of 5% or 1% is typically used. We then look in the standard t–distribution table, available in statistics textbooks. This table is normally arranged by degrees of freedom as rows and significance levels as columns. Suppose we want to make sure whether the difference between M_1 and M_2 is significantly different for 95% of the population, that is, $sig = 5\%$ or 0.05. We ought to find the t–distribution value related to $k - 1$ degrees of freedom (or 9 degrees of freedom for our example) from the table. However, because the t–distribution is symmetric, typically only the top percentage points of the distribution are shown. Therefore, we look up the table value for $z = sig/2$, which in this case, is 0.025, where z is also referred to as a confidence limit. If $t > z$ or $t < -z$, then our value of t lies in the rejection region, within the distribution's tails. This means that we can reject the null hypothesis that the means of M_1 and M_2 are the same and conclude that there is a statistically significant difference between the two models. Otherwise, if we cannot reject the null hypothesis, we find that any difference between M_1 and M_2 can be attributed to chance [73].

CHAPTER 4
Proposed Frameworks

As mentioned before, one major constraint of previous automatic systems is their need for high–quality images, which is not practical, especially when working with a lower resolution cell phone captured images with not much of perfect lighting and uniformity in the setting. Our proposed methods overcome this shortcoming by grayscaling and downsizing reference RGB images to 128 × 128 grayscale images to perform feature extraction easier and faster.

The previous automatic systems extracted geometrical features that need several preprocessing steps using different image filtering and morphological operations like lens distortion correction, background removal, specimen's body edge identification, clipping the legs and antennas from the image, and smoothing the clipped edge before feature extraction. We minimized the preprocessing to only background elimination operation.

Lastly, the accuracy of 80–88% of currently developed methods leaves room for improvement. We improved accuracy by utilizing different state–of–the–art feature extraction and classification algorithms. Instead of extracting ten geometrical features, we used PCA to extract the best 50 and 150 scattered variance–based features from images. We also applied feature selection and oversampling techniques before classification. For classification, we utilized and personalized DT, RF and SVM mining algorithms. We also applied deep neural networks, which does not need the feature extraction and feature selection phases before classification.

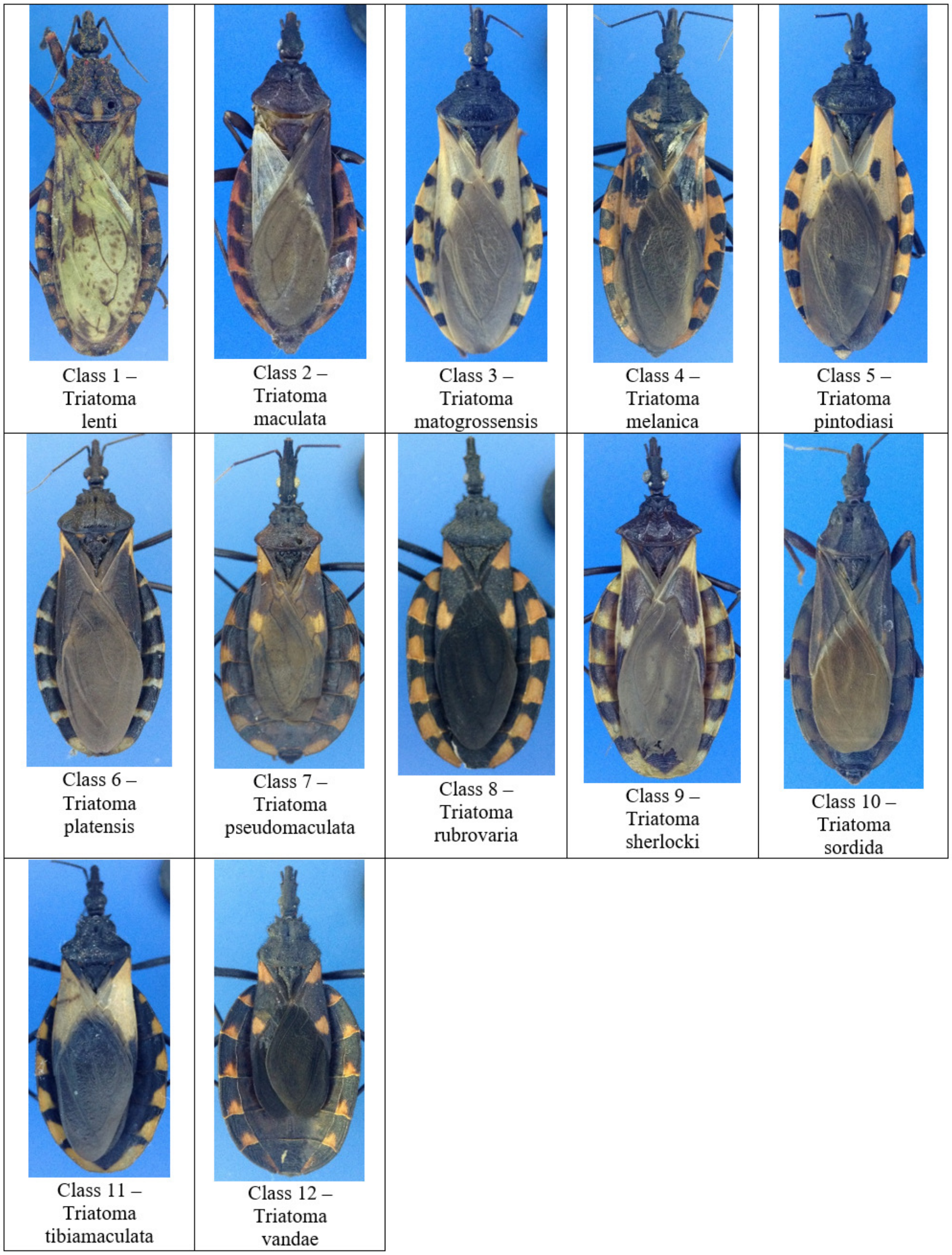
We used 10–fold cross–validation in training and testing steps of our classifiers to get robust unbiased results. We evaluated our results by reporting precision, recall, and f–measure identification factors. We also performed a paired t–test exam to assess the statistical independence of classification models. In the following sections, these steps, along with the dataset description, are explained in detail.

4.1 DATASET

As mentined, because of the nature of kissing bugs and limitations in gathering many benign and infectious vectors, there is a shortage of sufficient sets of images in this field. The available ones are rather small datasets. Due to this shortage, in this research, we used the same dataset of images provided by Gurgel et al. [50] consisting of 2030 insect photos; to be able to compare the accuracy rates. Brazilian species belong to 6 different families of Triatomine bugs called Cavernicola, Eratyrus, Panstrongylus, Psammolestes, Rhodnius, and Triatoma; and Mexican species belong to 2 different families called Panstrongylus and Triatoma. The overall number of species is 51 classes composed of 39 Brazilian and 12 Mexican species. The number of images varies from 13 to 89 per class. The total number of 2030 images consisting of 1620 Brazilian and 410 Mexican species. In Figure 4.1 one vector sample from each 12 Mexican classes and in Figures 4.2 and 4.3 one vector sample from each 39 Brazilian classes are shown.

4.2 PREPROCESSING

Several preprocessing steps of reference methods are reduced to one background removal step to reduce computational complexity and time consumption of this phase. After that, by grayscaling and downsizing the image we train our system to work with low–resolution, low–quality images and make it robust to this weakness. First, the background (blue background color along with size standard white circle) is separated from the foreground (insect's body) using image segmentation K–means clustering method in Matlab and foreground is contoured to define a rectangle around bug's body, and the image is cropped afterward. In the next step, the image is grayscaled and downsized from 1936×2592 RGB images to 128×128 grayscale images. Preprocessing steps and results for one Brazilian vector are shown in Figure 4.4.



Class 1 – Triatoma lenti

Class 2 – Triatoma maculata

Class 3 – Triatoma matogrossensis

Class 4 – Triatoma melanica

Class 5 – Triatoma pintodiasi

Class 6 – Triatoma platensis

Class 7 – Triatoma pseudomaculata

Class 8 – Triatoma rubrovaria

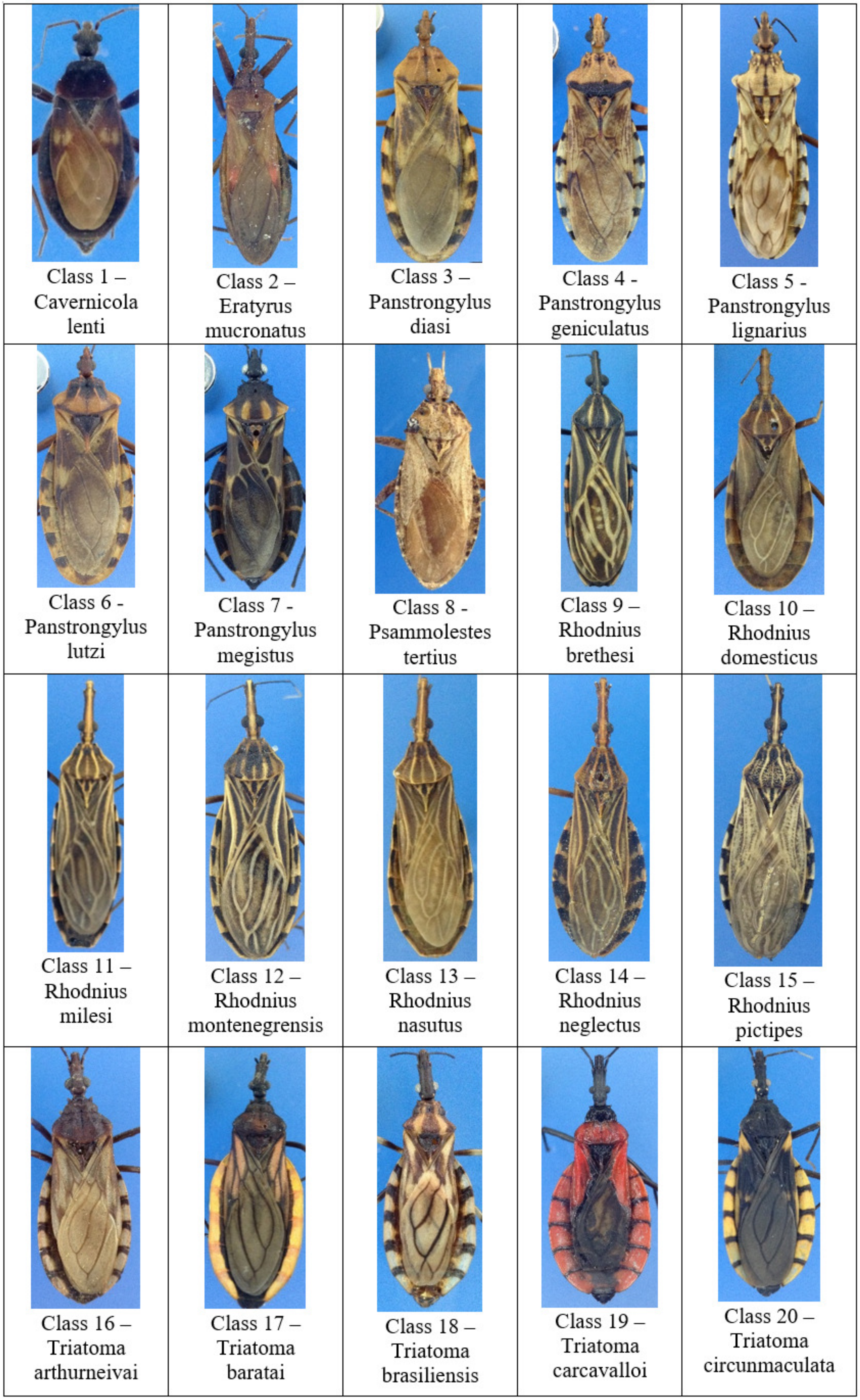
Class 9 – Triatoma sherlocki

Class 10 – Triatoma sordida

Class 11 – Triatoma tibiamaculata

Class 12 – Triatoma vandae

Figure 4.1: Mexican dataset samples. 12 classes of Panstrongylus and Triatoma bugs.



Class 1 – Cavernicola lenti

Class 2 – Eratyrus mucronatus

Class 3 – Panstrongylus diasi

Class 4 - Panstrongylus geniculatus

Class 5 - Panstrongylus lignarius

Class 6 - Panstrongylus lutzi

Class 7 - Panstrongylus megistus

Class 8 - Psammolestes tertius

Class 9 – Rhodnius brethesi

Class 10 – Rhodnius domesticus

Class 11 – Rhodnius milesi

Class 12 – Rhodnius montenegrensis

Class 13 – Rhodnius nasutus

Class 14 – Rhodnius neglectus

Class 15 – Rhodnius pictipes

Class 16 – Triatoma arthurneivai

Class 17 – Triatoma baratai

Class 18 – Triatoma brasiliensis

Class 19 – Triatoma carcavalloi

Class 20 – Triatoma circunmaculata

FIGURE 4.2: Part 1 of Brazilian dataset samples. 20 classes consisting of Cavernicola, Eratyrus, Panstrongylus, Psammolestes, Rhodnius, and Triatoma bugs.

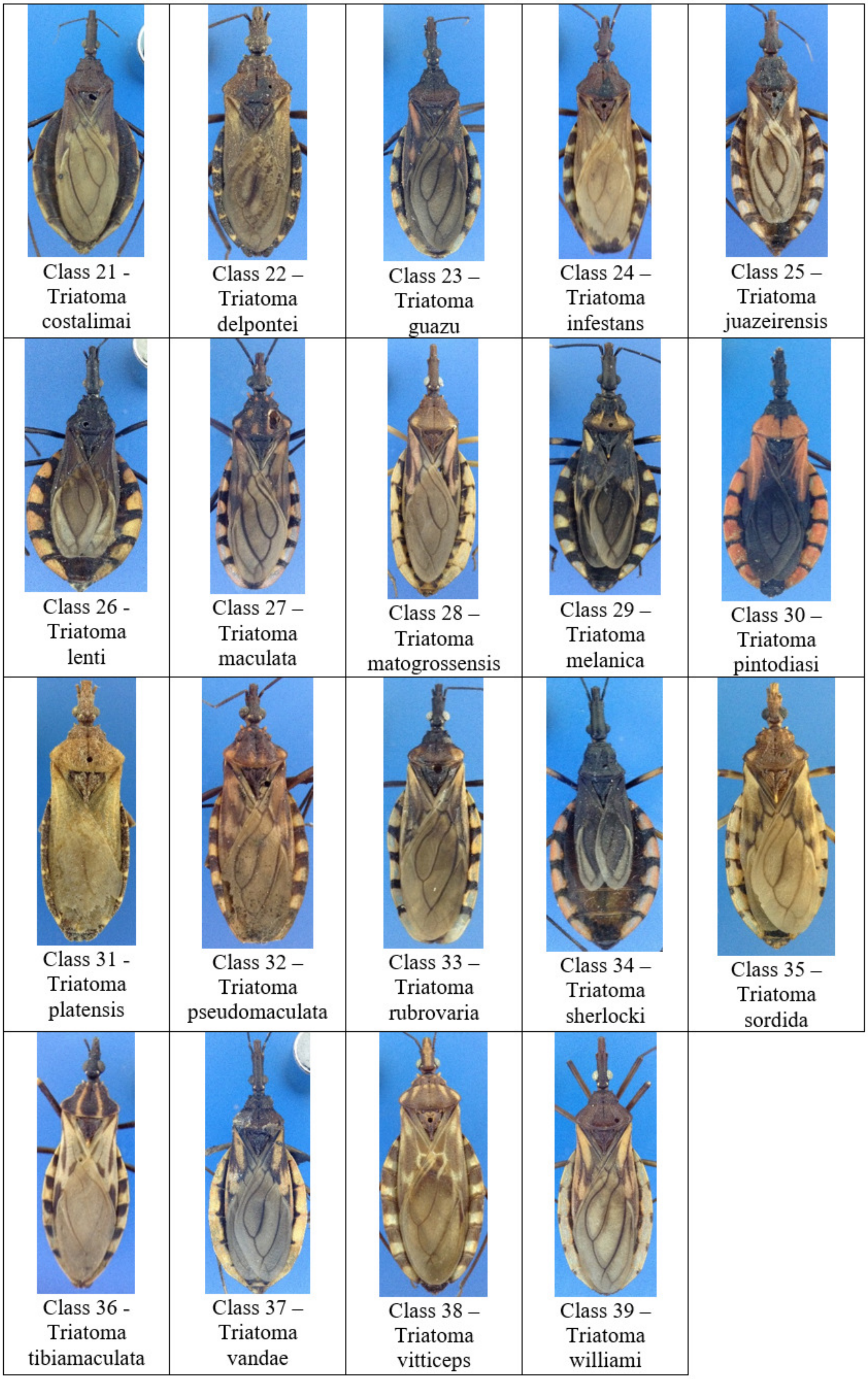

Class 21 - Triatoma costalimai
Class 22 – Triatoma delpontei
Class 23 – Triatoma guazu
Class 24 – Triatoma infestans
Class 25 – Triatoma juazeirensis
Class 26 - Triatoma lenti
Class 27 – Triatoma maculata
Class 28 – Triatoma matogrossensis
Class 29 – Triatoma melanica
Class 30 – Triatoma pintodiasi
Class 31 - Triatoma platensis
Class 32 – Triatoma pseudomaculata
Class 33 – Triatoma rubrovaria
Class 34 – Triatoma sherlocki
Class 35 – Triatoma sordida
Class 36 - Triatoma tibiamaculata
Class 37 – Triatoma vandae
Class 38 – Triatoma vitticeps
Class 39 – Triatoma williami

Figure 4.3: Part 2 of Brazilan Dataset. 19 classes of Triatoma bugs.

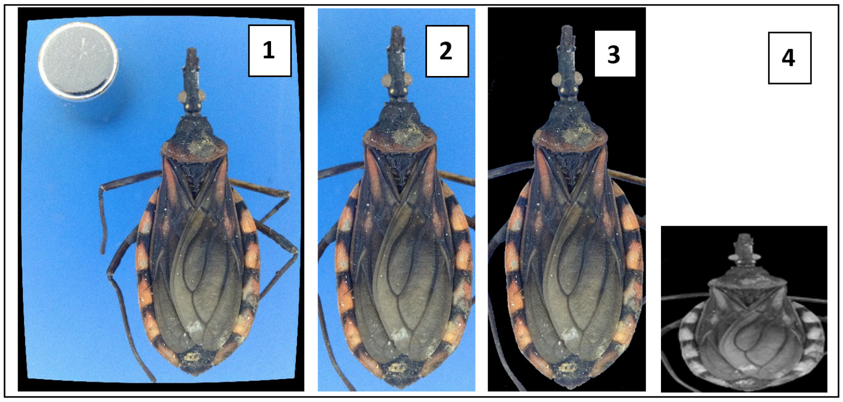


Figure 4.4: Summary of preprocessing Steps for an example Brazilian image of Triatoma guazu (1) Raw image (2) cropped image (3) background removed image (4) gray–scaled down–sized image

4.3 FEATURE EXTRACTION

We utilized PCA for feature extraction. In two different experiments, 50 and 150 features were extracted using the PCA algorithm. After rearranging $[128 \times 128]$ grayscale images into vectors of $16,384$ elements, we stacked them in a matrix named "imageVectors"of size $[1620 \times (16384 + 1)]$ (the last column is for class label). Then imageVectors matrix was normalized using the following method (Table 4.1):

Table 4.1: Max/Min normalization before applying PCA.

Step 1: min = overall minimum of imageVectors matrix
Step 2: max = overall maximum of imageVectors matrix
Step 3: normalized–Matrix = (imageVectors - min)/(max - min)

After that, the normalized–matrix is fed to PCA. The covariance matrix is computed and eigenvector and eigenvalues are extracted from the conaviance matrix on descending order. In the first experiment, we selected a featureVector of 50 eigenvectors corresponding to 50 largest eigenvalues, and in the second experiment we selected a featureVector of 150 eigenvectors corresponding to 150 largest eigenvalues. The two feature vectors were multiplied by the normalized data matrix and exported as PCAData50 and PCAData150, respectively. We then normalized feature vectors using the same max/min normalization method (Table 4.1) to prepare it for the classification phase. Since the next steps are performed in Weka, Matlab matrix files (i.e., ".mat" files) are converted to ARFF files suitable for processing in Weka. We then utilize the Weka software for the rest of the steps.

4.4 FEATURE SELECTION

We applied Weka's attribute selection filter on ARFF files to optimize the number of features for Mexican and Brazilian datasets separately before applying classification algorithms. We chose "CfsSubsetEvaluator" as the attribute evaluator algorithm and "BestFirst" algorithm as the search method. CfsSubsetEvaluator evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features that are highly correlated with the class while having low intercorrelation are preferred. BestFirst algorithm searches the space of attribute subsets by greedy hillclimbing augmented with a backtracking facility.

With trial and error, the optimal configuration values for *CfsSubsetEvaluator* (*poolSize*[3] = 8 and *numThreads* = 8, $\geq$ size of thread pool) and BestFirst (*direction* = forward and searchTermination[4]= 5) search method are achieved and applied.

4.5 DATA BALANCING

We oversampled our dataset using Weka "Resample" technique with proper configuration (*noReplacement* = *false*, *biasToUniformClass* = 1.0, and *sampleSizePercent* = 200). After oversampling we balanced our feature datasets as much as possible using Weka "ClassBalancer" filter with number of discretization *intervals* = 20.

ARFF data file sizes are almost doubled up from 1620 rows to 3237 rows for Brazilian and from 410 rows to 816 rows for Mexican datasets using oversampling data technique; and the 150–features are reduced to 125 for Brazilian, and 132 for

[3] The size of the thread pool, for example, the number of cores in the CPU

[4] specifies the number of consecutive non–improving nodes to allow before terminating the search

Mexican species; and the 50–features are reduced to 29 for Brazilian and 32 for Mexican datasets using attribute Selection technique. After the feature selection and data balancing steps, classification algorithms were applied to the processed feature datasets.

4.6 CLASSIFICATION

We utilized DT as our test base algorithm for classification phase. We compared other classification algorithms' results with Weka implementation of DT called J48. We also utilized the Weka implelemtation of RF and Compare our acquired results of DT with it. The third utilized classification algorithm is Weka implementation of SVM for multiclasses (more than two class) called SMO.

After feature extraction and data balancing steps, the dataset is ready for the classification stage. We applied three different supervised classifications of DT (called approach 1), RF (called approach 2), and SVM (called approach 3), which shaped three different approaches in our research. Both sets of features (29 and 125 for Brazilian and 32 and 132 for Mexican) are fed to each algorithm, and the results are reported, compared, and evaluated. The flowchart of our three different classification approaches is depicted in Figure 4.5.

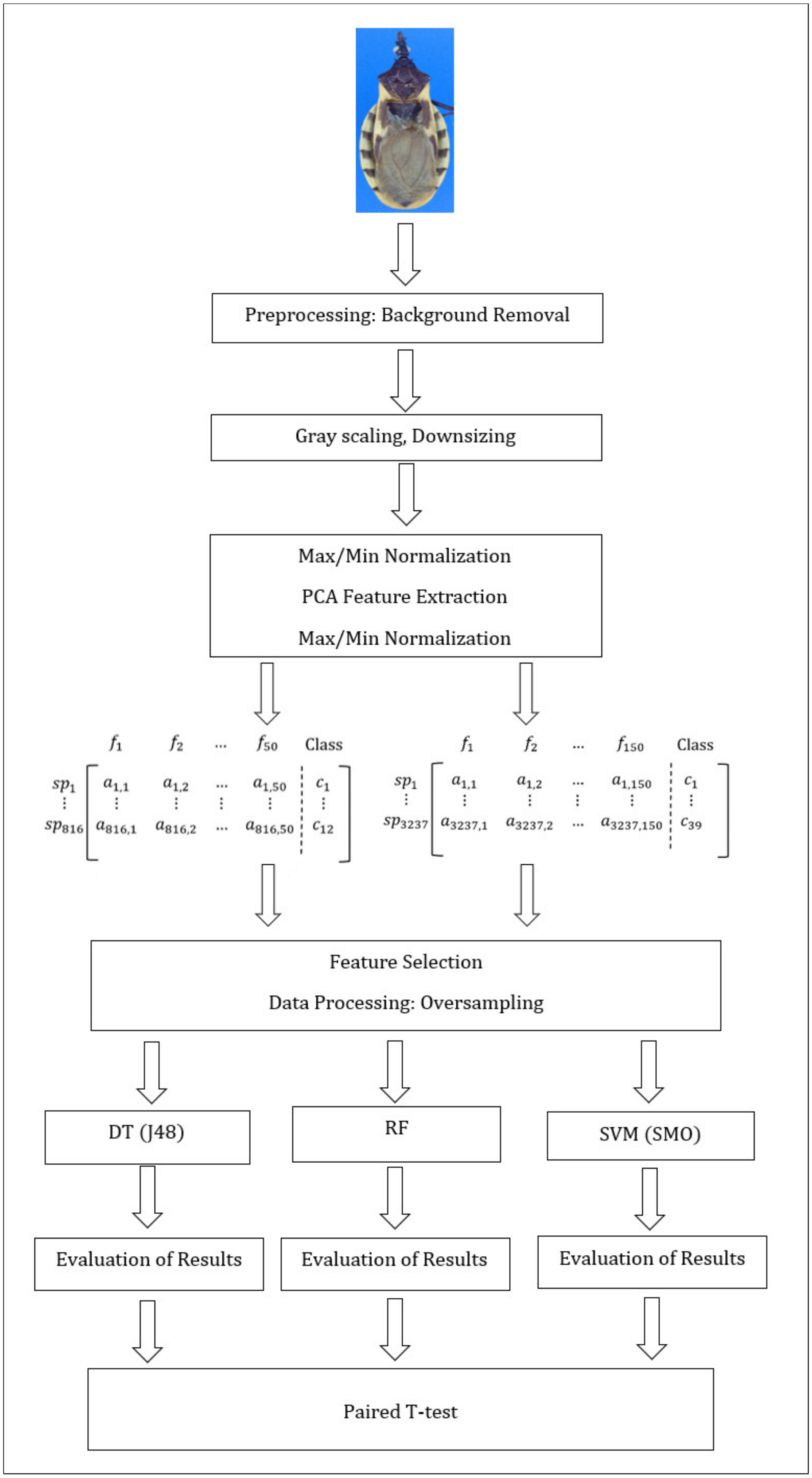

Preprocessing: Background Removal
Gray scaling, Downsizing
Max/Min Normalization
PCA Feature Extraction
Max/Min Normalization
f_1 f_2 ... f_{50} Class
sp_1 $a_{1,1}$ $a_{1,2}$ ... $a_{1,50}$ c_1
sp_{816} $a_{816,1}$ $a_{816,2}$ ... $a_{816,50}$ c_{12}
f_1 f_2 ... f_{150} Class
sp_1 $a_{1,1}$ $a_{1,2}$ ... $a_{1,150}$ c_1
sp_{3237} $a_{3237,1}$ $a_{3237,2}$ ... $a_{3237,150}$ c_{39}
Feature Selection
Data Processing: Oversampling
DT (J48)
RF
SVM (SMO)
Evaluation of Results
Evaluation of Results
Evaluation of Results
Paired T-test

FIGURE 4.5: Framework of our three different data mining based approaches for classification. PCA+DT, PCA+RF, and PCA+SVM for both 50 and 150 feature sets.

We also implemented two deep learning based methods. First method has the same neural network structure as VGG16. We utilized ImageNet as the checkpoint and fined-tune VGG16 network with our own dataset. As you will see in the next section, the performance of VGG16 is not as high as we expected it to be (we were seeking comparable results with other proposed approaches) due to small dataset dilemma.

Therefore we designed a 7-layer CNN with the same architecture as VGG16 with less and smaller convolution and fully connected layers. Our 7-layer CNN consists of 5 convolution layers, 2 fully connected layers, and 1 Softmax output layer. Since the network is rather small, we trained it from scratch using our own image dataset. Relu is utilized as the activation function and max-pooling is performed on pooling layers. The 7-layer CNN structure is finalized after designing and testing different architectures with different sizes and chose the one with the best test accuracy as our proposed deep neural network. The design and architecture map of VGG16 along with our proposed 7-layer CNN are depicted in Figures 4.6 and 4.7, respectively. As you can see, 7-layer CNN is smaller version of VGG16 with fewer and smaller convolution and fully connected layers.

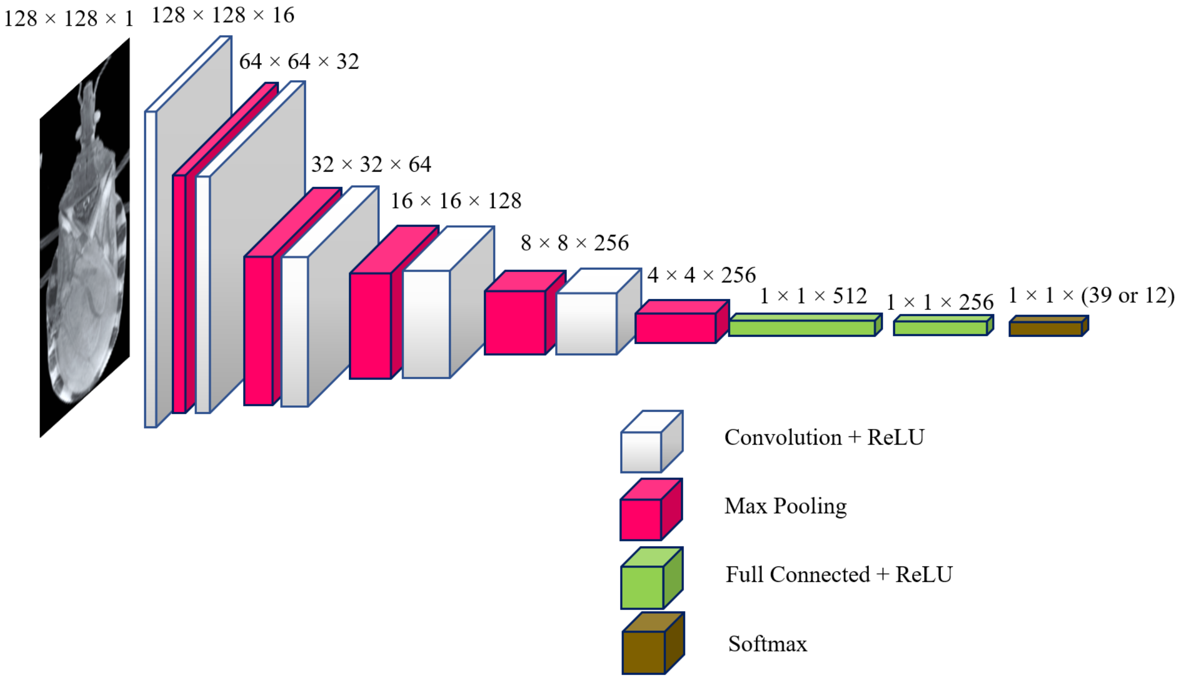


FIGURE 4.6: Our optimal design of deep neural network

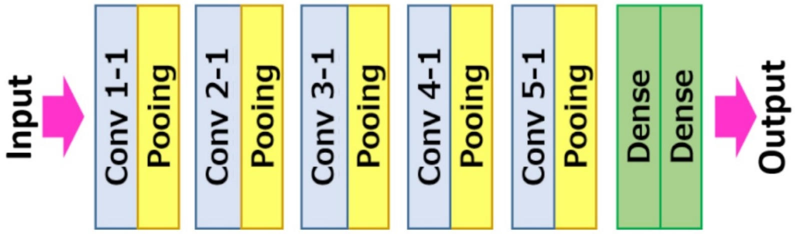


FIGURE 4.7: Our optimal designed deep neural network – architecture map

As you can see in the desing of our CNN, after the first convolution layer, the size of featuremap is the same as the original image. "Valid" convolution with zero pading is applied to mantain the same size of the image after the first convolution. Pooling is applied with factor 2. Therefore the first pooling shrinks the featuremap size by factor 2. Same convolution and pooling operations are applied at next layers. The twpo fully connected layers have 512 and 256 outputs respectively. Finally, the softmax layer has 256 inputs and 39 or 12 outputs for Brazilian and Mexican classes respectively.

CHAPTER 5
Results and Discussion

As described in Chapter 4, four different approaches are designed and implemented for the automatic identification of CD Vectors in this research: approach 1: PCA+DT, approach 2: PCA+RF, approach 3: PCA+SVM, and finally approach 4: CNNs.

In the first section, the results of the first approach (PCA+DT) in the cases of involving or excluding feature selection step is discussed. Then the detailed evaluation of results and discussion for the first three approaches are explained in the following three sections.

Weka is utilized for the first data mining–based classification algorithms. When evaluating multi–class classification models, Weka outputs a weighted average of the per–class precision, recall, and f–measure: it computes these statistics for each class individually, treating the corresponding class as the "positive" class and the union of the other classes as the negative class, and computes a weighted average of these per–class statistics, with a per–class weight that is equal to the proportion of data in that class. In a multi–class classification problem (including two–class ones) in Weka, micro–averaged precision, recall and f–measure are all almost the same and identical to classification accuracy as measured by the percentage of correctly classified instances. Therefore, since f–measure is the combination of precision and recall values, it would also be considered and reported as the accuracy value of the classifier in our experiments.

After first evaluation report for the first three approaches, the paired t–test applied on these three approaches is described in Section 5.5. Approach 4: CNNs is explained in Section 5.6. Finally, the overall comparison of all the approaches, along with previously developed methods, is explained in Section 5.7.

5.1 FEATURE SELECTION

PCA algorithm extracts orthogonal features. If the optimal number of eigenvector is extracted, the features would be completely independant and no data redundancy will be available in the featureSet. On the other hand, finding the optimal number of PCA features is a delicate and rather complicated task. There are techniques such as scree plot and finding the elbow of the diagram to determine the optimal number of PCA features, but the diagram has different elbows and determining which elbow is the best one to select is a hard goal to achieve. Therefore, we applied trial and error to find the optimal number of PCA features. In two different experiments, the total of 50 and 150 eigenvectors are extracted to represent 90% and 99% of the values of the normalized sorted eigenvalues for both Brazilian and Mexican species. Then, a correlation based feature selection technique is applied on these featureSets to find the optimal subset of features for both 50 and 150 featureSets. Total number of 32 and 132 features are selected for Mexican species. Total number of 29 and 125 features are selected for Brazilian species. Both groups of selected featureSets are fed to our approach 1: PCA+DT algorithm and the accuracies are compared. The results showed that featureSets after feature selection had higher accuracy (more than 5%) than the featureSets before feature selection step. Based on these results, we decided to apply feature selection step as a necessary step in our three data mining based approaches (approach 1 to approach 3).

5.2 APPROACH 1: PCA+DT

In the following tables (Tables 5.1 and 5.2), the percentage values of precision, recall, f–measure and accuracy for every class of Brazilian and Mexican species along with the number of species per class for two different sets of features (50 and 150) are depicted for the PCA+DT method. The values for parameters of DT are set to a *confidence factor* of 0.25 and a *batch size* of 200.

As we can see, the results of 150–features are better than 50–features for both the Brazilian and Mexican datasets. On the other hand, extracting 150–features is more time consuming than extracting 50–features. So, there is a trade–off between time consumption and the desired accuracy. The results of 150–features are perfect (100%) for both Brazilian and Mexican classes. Because of the close infrastructure of the DT and RF algorithms, a paired t–test is performed on them to examine the statistical independence of these two algorithms.

TABLE 5.1: Summary of species analyzed, sample sizes of photographs, and accuracy for the 12 species of Mexican triatomine bugs analyzed in this study.

PCA+DT		**# of features**							
Mexican Species		50 → 32				150 → 132			
Species	**Sp #**	**prec**	**rec**	**f–meas**	**accu**	**prec**	**rec**	**f–meas**	**accu**
Class 1	13	100	100	100	100	100	100	100	100
Class 2	31	100	100	100	100	100	100	100	100
Class 3	44	100	100	100	100	100	100	100	100
Class 4	30	100	100	100	100	100	100	100	100
Class 5	40	98.60	100	99.30	99.30	100	100	100	100
Class 6	12	97.10	100	98.60	98.60	100	100	100	100
Class 7	52	100	98.50	99.30	99.30	100	100	100	100
Class 8	23	100	100	100	100	100	100	100	100
Class 9	46	100	100	100	100	100	100	100	100
Class 10	16	100	100	100	100	100	100	100	100
Class 11	43	98.50	100	99.30	99.30	100	100	100	100
Class 12	60	100	95.60	97.70	97.70	100	100	100	100
Weighted Avg	410	99.50	99.50	99.50	99.50	100	100	100	100

In the Brazilian results with 50–features, the insects of the two classes of 13 and 14 which have lower accuracy rates compared to other classes, have close characteristics (Type and appearance of the vector). Therefore, even though the number of bugs is rather large in these two classes, they are misclassified by one another which resulted in lower classification rate for both. The same scenario happened for pairs in 17 and

24 classes, and pairs in 17 and 32 classes. This dilemma is significantly improved in the 150–features case where more distinguishing features are extracted.

5.3 APPROACH 2: PCA+RF

In the following tables (Tables 5.3 and 5.4), the percentage values of precision, recall, f–measure and accuracy for every class of Brazilian and Mexican species along with the number of species per class and two different sets of features (50 and 150) are depicted for PCA+RF method. The assigned values of parameters for the RF classification algorithm: *batch size* of 200, *number of iterations* of 500, and *number of execution slots* of 8 (same as the number of CPU cores in the system).

As you can see, 50 and 150 features' accuracy rates are perfect 100% for Mexican species.

For the Brazilian dataset, accuracy of 150–features is perfect 100%, while the accuracy of 50–features is 98.80%. Even though the results of 50–features is slightly less accurate than 150–features, it is still acceptable and promising values for our purpose. Brazilian dataset has more classes and is more complicated than the Mexican dataset. Therefore, it is reasonable to observe slightly less accurate for it.

Table 5.2: Summary of species analyzed, sample sizes of photographs, and accuracy for the 39 species of Brazilian triatomine bugs analyzed in this study.

PCA+DT		**# of features**							
Brazilian Species		50 → 29				150 → 125			
Species	**Sp #**	**prec**	**rec**	**f–meas**	**accu**	**prec**	**rec**	**f–meas**	**accu**
Class 1	13	98.80	100	99.40	99.40	100	100	100	100
Class 2	31	100	100	100	100	100	100	100	100
Class 3	44	98.80	100	99.40	99.40	100	100	100	100
Class 4	30	100	100	100	100	100	100	100	100
Class 5	40	98.80	100	99.40	99.40	100	100	100	100
Class 6	12	100	100	100	100	100	100	100	100
Class 7	52	80.50	74.70	77.50	77.50	100	100	100	100
Class 8	23	96.50	100	98.20	98.20	100	100	100	100
Class 9	46	100	100	100	100	100	100	100	100
Class 10	16	98.80	100	99.40	99.40	100	100	100	100
Class 11	43	100	100	100	100	100	100	100	100
Class 12	60	100	100	100	100	100	100	100	100
Class 13	80	81.70	69.9	75.30	75.30	100	100	100	100
Class 14	68	90.70	81.90	86.10	86.10	100	100	100	100
Class 15	57	96.20	90.40	93.20	93.20	100	100	100	100
Class 16	32	97.60	100	98.80	98.80	100	100	100	100
Class 17	29	96.50	100	98.20	98.20	100	100	100	100
Class 18	64	91.10	86.70	88.90	88.90	100	100	100	100
Class 19	38	100	100	100	100	100	100	100	100
Class 20	22	93.30	100	96.50	96.50	100	100	100	100
Class 21	64	83.50	79.50	81.50	81.50	100	100	100	100
Class 22	31	100	100	100	100	100	100	100	100
Class 23	29	96.50	100	98.20	98.20	100	100	100	100
Class 24	55	92.80	92.80	92.80	92.80	100	100	100	100
Class 25	21	100	100	100	100	100	100	100	100
Class 26	40	98.80	100	99.40	99.40	100	100	100	100
Class 27	40	98.80	100	99.40	99.40	100	100	100	100
Class 28	33	100	100	100	100	100	100	100	100
Class 29	29	97.60	100	98.80	98.80	100	100	100	100
Class 30	26	100	100	100	100	100	100	100	100
Class 31	28	100	100	100	100	100	100	100	100
Class 32	56	82.40	84.30	83.30	83.30	100	100	100	100
Class 33	55	84.40	91.60	87.90	87.90	100	100	100	100
Class 34	32	96.50	100	98.20	98.20	100	100	100	100
Class 35	104	75.00	75.90	75.40	75.40	100	100	100	100
Class 36	41	98.80	100	99.40	99.40	100	100	100	100
Class 37	30	100	100	100	100	100	100	100	100
Class 38	48	100	100	100	100	100	100	100	100
Class 39	20	100	100	100	100	100	100	100	100
Weighted Avg	1620	95.50	95.60	95.50	95.50	100	100	100	100

Table 5.3: Summary of species analyzed, sample size of images, and accuracy for the 12 species of Mexican triatomine bugs analyzed in this study.

PCA+RF		**# of features**							
Mexican Species		50 → 32				150 → 132			
Species	**Sp #**	**prec**	**rec**	**f–meas**	**accu**	**prec**	**rec**	**f–meas**	**accu**
Class 1	13	100	100	100	100	100	100	100	100
Class 2	31	100	100	100	100	100	100	100	100
Class 3	44	100	100	100	100	100	100	100	100
Class 4	30	100	100	100	100	100	100	100	100
Class 5	40	100	100	100	100	100	100	100	100
Class 6	12	100	100	100	100	100	100	100	100
Class 7	52	100	100	100	100	100	100	100	100
Class 8	23	100	100	100	100	100	100	100	100
Class 9	46	100	100	100	100	100	100	100	100
Class 10	16	100	100	100	100	100	100	100	100
Class 11	43	100	100	100	100	100	100	100	100
Class 12	60	100	100	100	100	100	100	100	100
Weighted Avg	410	100	100	100	100	100	100	100	100

In the Brazilian results with 50–features the insects of the two classes of 13 and 14, which have lower accuracy compared to other classes, have close characteristics, and their insects are misclassified by one another, which resulted in lower classification rate for both. The same scenario happened for pairs in 21 and 32 classes. This dilemma is resolved by extracting more distinguishing features (150–features) from images.

Table 5.4: Summary of species analyzed, sample sizes of photographs, and accuracy for the 39 species of Brazilian triatomine bugs analyzed in this study.

PCA+RF		**# of features**							
Brazilian Species		50 → 29				150 → 125			
Species	**Sp #**	**prec**	**rec**	**f–meas**	**accu**	**prec**	**rec**	**f–meas**	**accu**
Class 1	13	100	100	100	100	100	100	100	100
Class 2	31	100	100	100	100	100	100	100	100
Class 3	44	100	100	100	100	100	100	100	100
Class 4	30	100	100	100	100	100	100	100	100
Class 5	40	100	100	100	100	100	100	100	100
Class 6	12	100	100	100	100	100	100	100	100
Class 7	52	92.90	95.20	94.00	94.00	100	100	100	100
Class 8	23	100	100	100	100	100	100	100	100
Class 9	46	100	100	100	100	100	100	100	100
Class 10	16	100	100	100	100	100	100	100	100
Class 11	43	100	100	100	100	100	100	100	100
Class 12	60	100	100	100	100	100	100	100	100
Class 13	80	96.00	86.70	91.10	91.10	100	100	100	100
Class 14	68	87.90	96.40	92.00	92.00	100	100	100	100
Class 15	57	100	100	100	100	100	100	100	100
Class 16	32	100	100	100	100	100	100	100	100
Class 17	29	100	100	100	100	100	100	100	100
Class 18	64	100	98.80	99.40	100	100	100	100	100
Class 19	38	100	100	100	100	100	100	100	100
Class 20	22	100	100	100	100	100	100	100	100
Class 21	64	93.70	89.20	91.40	91.40	100	100	100	100
Class 22	31	100	100	100	100	100	100	100	100
Class 23	29	100	100	100	100	100	100	100	100
Class 24	55	98.80	100	99.40	99.40	100	100	100	100
Class 25	21	100	100	100	100	100	100	100	100
Class 26	40	100	100	100	100	100	100	100	100
Class 27	40	100	100	100	100	100	100	100	100
Class 28	33	100	100	100	100	100	100	100	100
Class 29	29	100	100	100	100	100	100	100	100
Class 30	26	100	100	100	100	100	100	100	100
Class 31	28	100	100	100	100	100	100	100	100
Class 32	56	89.70	94.00	91.80	91.80	100	100	100	100
Class 33	55	100	100	100	100	100	100	100	100
Class 34	32	100	100	100	100	100	100	100	100
Class 35	104	95.10	92.80	93.90	93.90	100	100	100	100
Class 36	41	100	100	100	100	100	100	100	100
Class 37	30	100	100	100	100	100	100	100	100
Class 38	48	100	100	100	100	100	100	100	100
Class 39	20	100	100	100	100	100	100	100	100
Weighted Avg	1620	98.80	98.80	98.80	98.80	100	100	100	100

5.4 APPROACH 3: PCA+SVM

In the following tables (Tables 5.5 and 5.6), the percentage values of precision, recall, f–measure and accuracy for every class of Brazilian and Mexican species along with the number of species per class and two different sets of features (50 and 150) are depicted for PCA+SVM method.

The optimal parameter values for the SMO classification algorithm for Mexican species with 150–features are the complexity parameter *C* equal to 100, RBFKernel with *gamma* equal to 0.2, and *batch size* equal to 200. The grid search algorithm optimally achieves *C* and *gamma* values in Weka with *XBase* and *YBase* of 10, *XMax*, and *YMax* of 3.0, and *XMin* and *YMin* of -3.0. Data also is standardized before applying the RBFKernel function. After finding the right range for gamma (0.1) by grid search, we tested more precise values using trial and error (reaching a value of 0.2).

Using the same gridSearch method to find optimal parameter values, the optimal parameter values for SMO classification algorithm for Mexican species with 50–features are c of 100, RBFKernel with *gamma* of 0.15, and *batch size* of 200. The same test is applied for Brazilian species with *c* of 100 and *gamma* of 0.2 for 150–features, and *c* of 100 and *gamma* of 1 for 50–features.

As we can see, the results of 150–features are better than 50–features for both Brazilian and Mexican datasets. SVM is a very powerful and robust algorithm while dealing with high dimensional data. The results improve with extracting more features but extracting more than 150–features would be very time–consuming. Having more data points may strongly result in a perfect accuracy for the SVM algorithm. Although, SVM's current results are also very promising and outperform the results of previously developed methods with significant improvement (more than 10%).

Table 5.5: Summary of species analyzed, sample sizes of photographs, and accuracy for the 12 species of Mexican triatomine bugs analyzed in this study.

PCA+SVM		**# of features**							
Mexican Species		50 → 32				150 → 132			
Species	**Sp #**	**prec**	**rec**	**f–meas**	**accu**	**prec**	**rec**	**f–meas**	**accu**
Class 1	13	100	100	100	100	100	100	100	100
Class 2	31	100	100	100	100	100	100	100	100
Class 3	44	82.60	83.80	83.20	83.20	98.50	98.50	98.50	98.50
Class 4	30	98.50	98.50	98.50 98.50		100	100	100	100
Class 5	40	100	98.50	99.30	99.30	100	100	100	100
Class 6	12	100	100	100	100	100	100	100	100
Class 7	52	100	100	100	100	100	100	100	100
Class 8	23	94.40	98.50	96.40	96.40	88.30	100	93.80	93.80
Class 9	46	97.00	94.10	95.50	95.50	100	86.80	92.90	92.90
Class 10	16	100	100	100	100	100	100	100	100
Class 11	43	100	100	100	100	100	100	100	100
Class 12	60	83.60	82.40	83.00	83.00	98.50	98.50	98.50	98.50
Weighted Avg	410	96.30	96.30	96.30	96.30	98.80	98.70	98.60	98.60

PCA+SVM faces the same dilemma as the two previous approaches (PCA+DT and PCA+RF) regarding the accuracy of similar pairs of classes. Even though the SVM results are less accurate than the results of DT and RF, SVM is a more powerful algorithm compared to the tree-based algorithms and more robust regarding overfitting. Being limited to a small dataset is a serious disadvantage for the performance of the SVM algorithm in this case.

Table 5.6: Summary of species analyzed, sample sizes of photographs, and accuracy for the 39 species of Brazilian triatomine bugs analyzed in this study.

PCA+SVM		**# of features**							
Brazilian Species		50 → 29				150 → 125			
Species	**Sp #**	**prec**	**rec**	**f–meas**	**accu**	**prec**	**rec**	**f–meas**	**accu**
Class 1	13	97.50	95.20	96.30	96.30	100	100	100	100
Class 2	31	100	100	100	100	100	100	100	100
Class 3	44	100	100	100	100	100	100	100	100
Class 4	30	79.10	86.70	82.80	82.80	100	100	100	100
Class 5	40	100	100	100	100	100	100	100	100
Class 6	12	96.20	90.40	93.20	93.20	100	97.60	98.80	98.80
Class 7	52	94.40	80.70	87.00	87.00	100	100	100	100
Class 8	23	98.80	97.60	98.20	98.20	100	100	100	100
Class 9	46	97.60	96.40	97.00	97.00	97.60	100	98.80	98.80
Class 10	16	90.20	100	94.90	94.90	80.40	98.80	88.60	88.60
Class 11	43	95.10	92.80	93.90	93.90	100	100	100	100
Class 12	60	82.80	92.80	87.50	87.50	96.30	95.20	95.80	95.80
Class 13	80	80.20	97.60	88.00	88.00	98.80	100	99.40	99.40
Class 14	68	100	91.60	95.60	95.60	100	98.80	99.40	99.40
Class 15	57	85.90	88.00	86.90	86.90	91.10	86.70	88.90	88.90
Class 16	32	94.30	100	97.10	97.10	100	100	100	100
Class 17	29	92.00	97.60	94.70	94.70	100	98.80	99.40	99.40
Class 18	64	100	97.60	98.80	98.80	100 100		100	100
Class 19	38	100	100	100	100	100	100	100	100
Class 20	22	100	100	100	100	100	100	100	100
Class 21	64	95.80	83.10	89.00	89.00	98.40	75.90	85.70	85.70
Class 22	31	96.30	95.20	95.80	95.80	95.20	96.40	95.80	95.80
Class 23	29	100	100	100	100	100	100	100	100
Class 24	55	97.60	100	98.80	98.80	100	100	100	100
Class 25	21	100	100	100	100	98.80	100	99.40	99.40
Class 26	40	87.50	84.30	85.90	85.90	87.40	91.60	89.40	89.40
Class 27	40	100	90.40	94.90	94.90	100 100		100	100
Class 28	33	100	100	100	100	100	100	100	100
Class 29	29	96.40	97.60	97.00	97.00	100	100	100	100
Class 30	26	100	100	100	100	100	100	100	100
Class 31	28	95.30	98.80	97.00	97.00	92.20	100	96.00	96.00
Class 32	56	92.90	95.20	94.00	94.00	100	100	100	100
Class 33	55	100	100	100	100	100	100	100	100
Class 34	32	94.00	95.20	94.60	94.60	100	97.60	98.80	98.80
Class 35	104	84.60	79.50	82.00	82.00	100	91.60	95.60	95.60
Class 36	41	100	91.60	95.60	95.60	100	100	100	100
Class 37	30	100	100	100	100	100	100	100	100
Class 38	48	100	100	100	100	100	100	100	100
Class 39	20	98.80	100	99.40	99.40	97.60	100	98.80	98.80
Weighted Avg	1620	95.50	95.30	95.30	95.30	98.30	98.20	98.20	98.20

5.5 PAIRED T–TEST

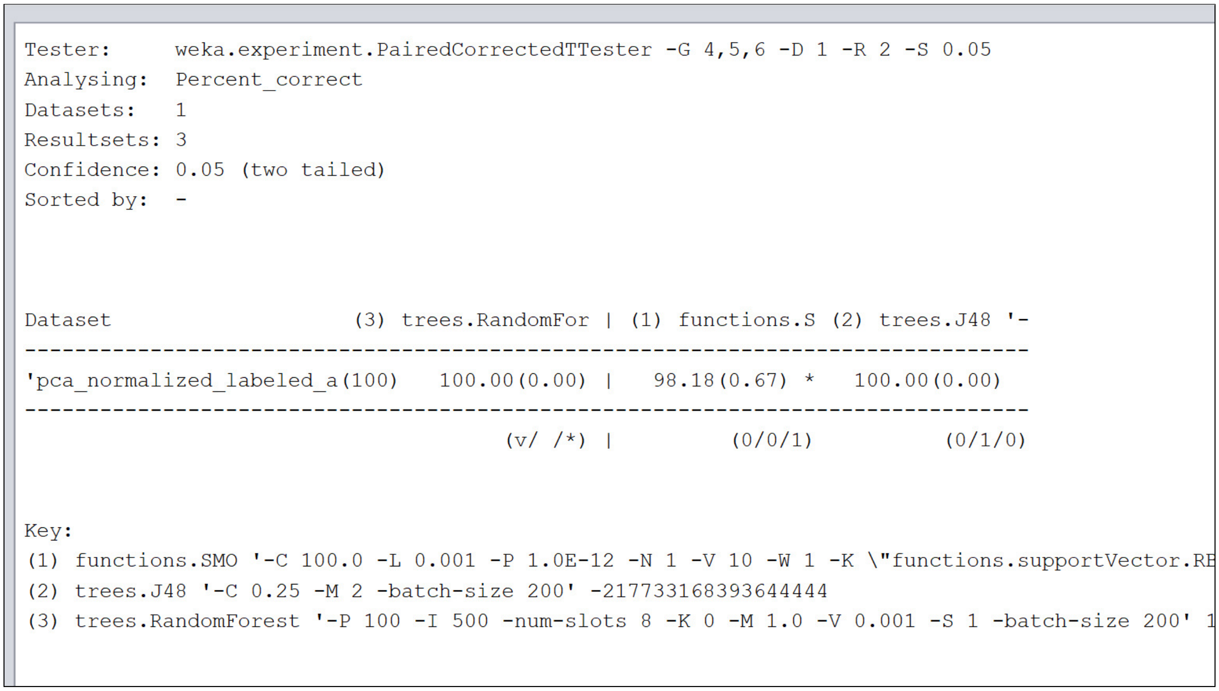
We performed Weka implementation of paired t–test for the first three approaches to test these algorithms' statistical independence of from each other. Since the results of 150-features outperform 50–features in all the experiments, we utilize this dataset for paired t–test for Brazilian and Mexican species.

We upload the 150–features Brazilian dataset with 3237 instances in the Weka Experimenter module. We choose a classification option (via regression) as the method and 10–fold cross–validation as the algorithms performance evaluator method. We add the three DT, RF, and SVM algorithms with optimal configurations in the algorithms section and set the number of repetitions of the test to 10. The evaluation is performed on each algorithm for $10 \times 10 = 100$ times with this setting. Then, we run the analyzer. After a considerable amount of time[5](100 runs of each algorithm = 300 runs) the results are ready for analyzing.

We utilized Weka software for performing paired t–test on our Weka based classification approaches. A paired t–test is implemented at the "Experimenter" module of Weka. After uploading our processed dataset in the experimenter module, we select 10–fold cross–validation as the evaluation metric to get robust, consistent results. The other factor in this process is the number of repetitions for the test. For example, by choosing ten as a number of repetitions, each algorithm will be performed $10 \times 10 = 100$ times for the sake of paired t–test evaluation. We also add our three algorithms of PCA+DT, PCA+RF, and PCA+SVM to the module and present and analyze paired t–test results in corresponding experimenter sections.

For Analyzing the results, we choose paired t–test at the testing method, "percent_correct" as the comparison field, significance level of 0.05, and Random Forest algorithm as the test base. The paired t–test for Mexican and Brazilian species are depicted in Figures 5.1 and 5.2, respectively.

[5] 15 minutes on HP laptop of Core i7 1.80 GHz with 16GRAM with Windows 10 operating System and Weka version of 3.8.4 – the needed heap size for Weka is also more than 10GBs for building all the models in this experiment

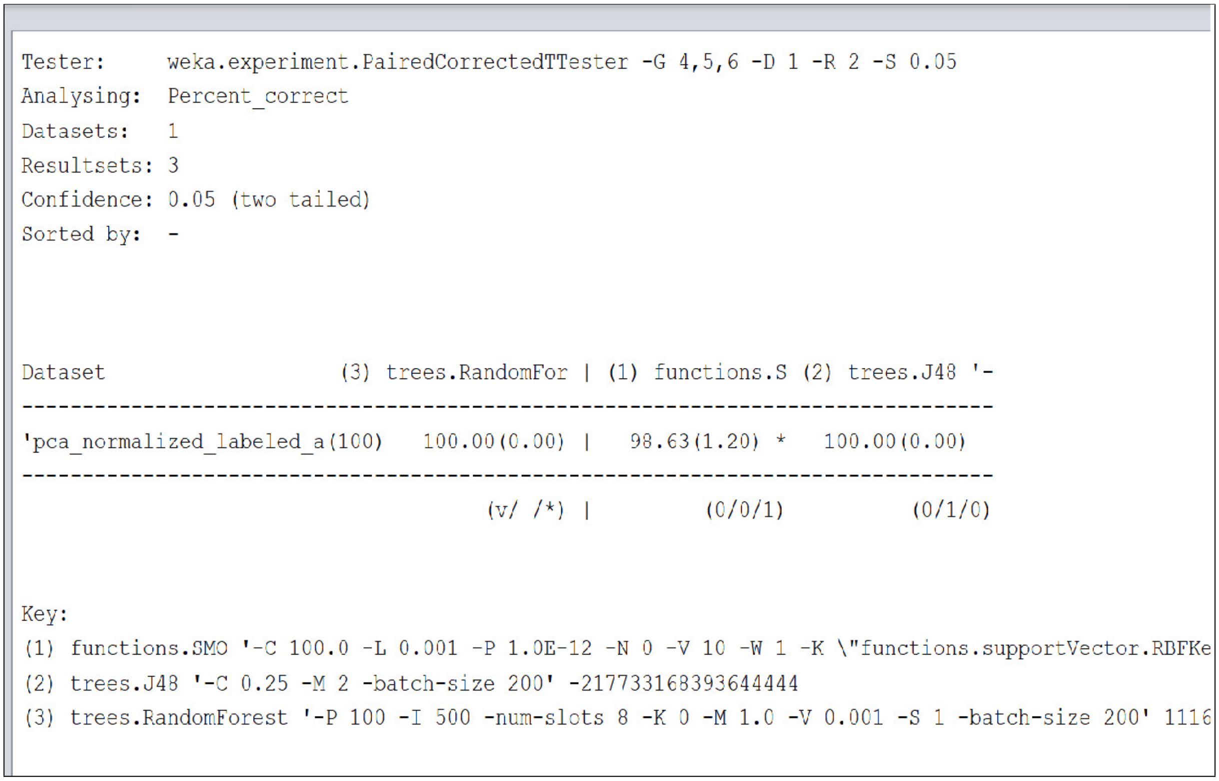


```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05
Analysing:  Percent_correct
Datasets:   1
Resultsets: 3
Confidence: 0.05 (two tailed)
Sorted by:  -



Dataset                   (3) trees.RandomFor | (1) functions.S (2) trees.J48 '-
--------------------------------------------------------------------------------
'pca_normalized_labeled_a(100)   100.00(0.00) |   98.18(0.67) *   100.00(0.00)
--------------------------------------------------------------------------------
                                      (v/ /*) |         (0/0/1)           (0/1/0)


Key:
(1) functions.SMO '-C 100.0 -L 0.001 -P 1.0E-12 -N 1 -V 10 -W 1 -K \"functions.supportVector.RE
(2) trees.J48 '-C 0.25 -M 2 -batch-size 200' -217733168393644444
(3) trees.RandomForest '-P 100 -I 500 -num-slots 8 -K 0 -M 1.0 -V 0.001 -S 1 -batch-size 200' 1
```

FIGURE 5.1: Paired t–test results comparsion for RF (Test base), DT and SVM for Brazilian species

As we can see, the output uses the annotation v or * to indicate that a specific result is statistically better (v) or worse (*) than the baseline scheme at the significance level specified (currently 0.05). Interpreting Brazilian results, the difference between DT and Rf is not statistically significant. But the results of SMO are worse than Rf, which makes sense based on its lower accuracy.

The process is the same for the Mexican dataset. We upload the 150–features Mexican dataset with 816 instances in the Weka Experimenter module, set the same configuration and run the test. We set the same configuration for analyzing the results as well. The interpretation of Mexican results is close to Brazilian results. The difference between DT and Rf is not statistically significant. But the SMO results are worse than Rf, which makes sense based on its lower accuracy rates. Because the difference between DT and RF algorithm is not statistically significant for both Brazilian and Mexican species only RF results are reported (RF outperformed DT in previous experiments) for comparison with other approaches.



```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05
Analysing:  Percent_correct
Datasets:   1
Resultsets: 3
Confidence: 0.05 (two tailed)
Sorted by:  -



Dataset                   (3) trees.RandomFor | (1) functions.S (2) trees.J48 '-
--------------------------------------------------------------------------------
'pca_normalized_labeled_a(100)   100.00(0.00) |   98.63(1.20) *   100.00(0.00)
--------------------------------------------------------------------------------
                                      (v/ /*) |         (0/0/1)          (0/1/0)


Key:
(1) functions.SMO '-C 100.0 -L 0.001 -P 1.0E-12 -N 0 -V 10 -W 1 -K \"functions.supportVector.RBFKe
(2) trees.J48 '-C 0.25 -M 2 -batch-size 200' -217733168393644444
(3) trees.RandomForest '-P 100 -I 500 -num-slots 8 -K 0 -M 1.0 -V 0.001 -S 1 -batch-size 200' 1116
```

Figure 5.2: Paired T–test results comparsion for RF (Test base), DT and SVM for Mexican species

5.6 Approach 4: Deep CNNs

As mentioned, two different structures are implemented and tested for deep learning classification. The first approach is designing VGG16 pre–trained with ImageNet dataset and fine–tuned with our dataset (Refer to Figure 3.8 for network structure). The second approach is creating a smaller 7–layer CNN and train it with our dataset from scratch (Refer to Figure 4.7 for network structure).

The network parameters for VGG16 are learning rate of 0.002, weight decay of 0.0005 and batch size of 64 and 128 for Mexican and Brazilian species respectively. The algorithm was run for 200 epochs. The network parameters for 7–layer CNN are learning rate of 0.0003, weight decay of 0.002 and batch size of 64 and 128 for Mexican and Brazilian species respectively. The network was run for 350 epochs. The accuracy rates of these two deep neural networks for Mexican and Brazilian datasets are shown in Table 5.7.

Table 5.7: Testing accuracy results comparsion for the two CNNs

Methods	Region	Precision	Recall	F1–score	Accuracy
VGG–16	**Brazil**	83.25	88.60	86.52	85.77
	Mexico	82.57	89.20	87.93	87.50
7–layer CNN	**Brazil**	95.33	97.57	96.82	96.93
	Mexico	96.71	96.48	96.53	96.26

[a] Values are (%) and average of all values for each country

[b] F1–score is the same factor as reported f–measure

Even though transfer learning (fine–tuning) is a very beneficial solution for accelerated and feasible convergence when dealing with rather small datasets, it does not work properly with our dataset (the dataset is too small to be able to fine–tune ImageNet weights properly). Therefore, it does not perform as well as we expected. The convergence is not optimal and accuracy rates are rather low. On the other hand, the results of smaller 7–layer CNN is acceptable and promising. Once the delimma of small dataset is overcome, better results will likely to be achieved.

5.7 Overall Results Comparison

In this section, we wrap up the results and discussion chapter by illustrating the complete comparison of the accuracy in the form of a bar chart. Based on our several experiments, we can infer some conclusions and filter out insignificant results from this chart. Firstly, the results of the 150–features dataset outperformed the results of the 50–features dataset in all the cases; therefore, we report only 150–feature results in the chart. Secondly, the difference between DT and RF is not statistically significant; therefore, only RF results are reported (RF outperformed DT). Lastly, between the two Deep NN algorithms, 7–layer CNN outperformed VGG16 significantly; therefore, only the 7–layer CNN results are reported. All results presented in this chapter are shown in Figure 5.3.

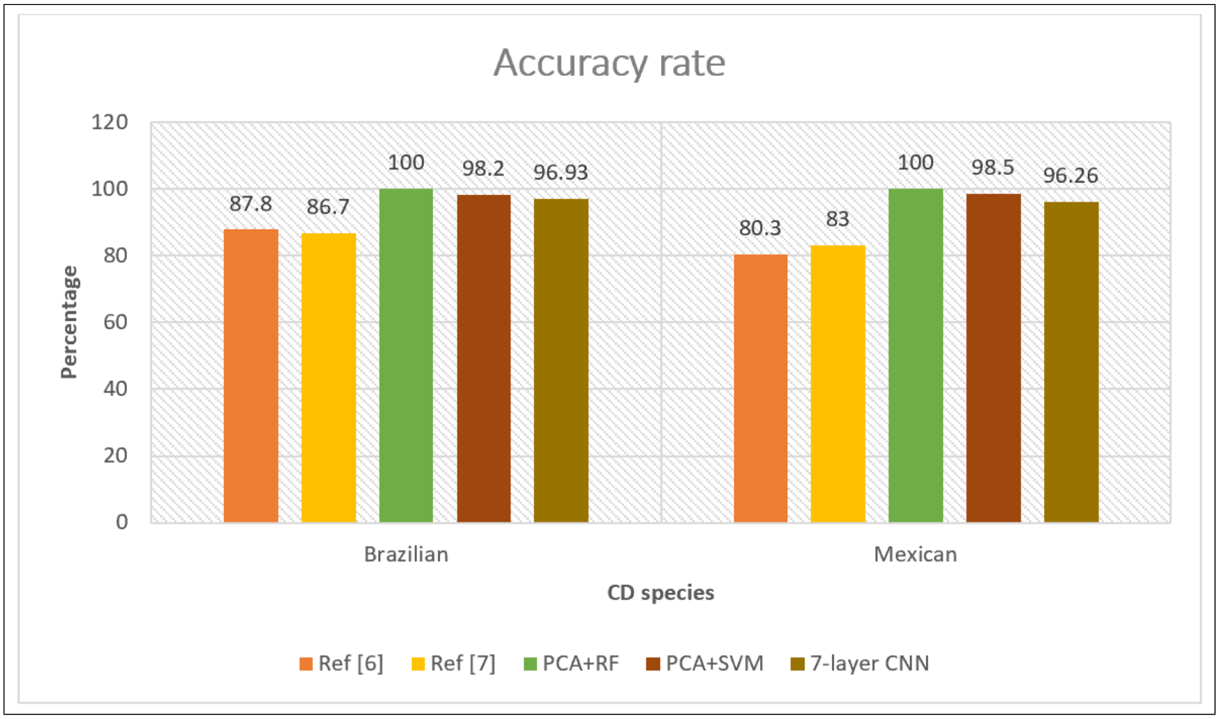


FIGURE 5.3: The comparison of the accuracy of our proposed approaches with previously developed systems.

As we can see, our proposed methods significantly outperformed previously developed systems. More than a 10% increase in accuracy in both Brazilian and Mexican species was demonstrated. Between our three PCA+RF, PCA+SVM, and 7–layer CNN algorithms, PCA+RF has perfect accuracy followed by PCA+SVM and 7–layer CNN. SVM and CNN are statistically better and more robust classification algorithms than RF and are less prone to overfitting. But in our research, the size of the image dataset was a limitation for the performance of these two state–of–the–art classification algorithms which emphasizes the challenge facing CD vector identification systems. We believe using larger image sets of kissing bugs will achieve better results using SVM and CNN algorithms.

Chapter 6
Conclusion

In this research, five different data mining–based and deep learning–based methods regarding CD vector identification are presented, and their results are compared with the two previously developed systems.

The first previously developed system consists of preprocessing, feature extraction, and classification steps. After several preprocessing steps, including lens distortion correction, background removal, specimen's body edge identification, clipping the legs and antennas from the image, and smoothing the clipped edge, they extracted ten geometrical features from images. For classification, they used a feed–forward neural network applied separately on Brazilian and Mexican species. They gathered a CD vector dataset consisting of more than 2000 vector images belonging to 51 different CD acquired species from Brazil and Mexico and performed their experiments using them as inputs. Their dataset is utilized in their second research paper (referred to as the second previously developed system) and used by our research. They achieved accuracy of 87.80% and 80.30% for Brazilian and Mexican species. The same research team designs the second previously developed system to improve the results of the first system. They used the same dataset in the second research as well. They implemented a deep neural network. The achieved accuracy for their techniques is 86.70% and 83.00% for Brazilian and Mexican species respectively. Based on these rates their second system outperforms their first designed system.

The two reference methods had three significant shortcomings that are addressed and improved in our research. Three out of five proposed algorithms have the same preprocessing steps as the first reference research with different sub–steps. We improved the overall results by enhancing each step of the reference algorithm.

The first dilemma is their need for high–resolution, high–quality images, which is not practical, especially when working with a low-resolution cell phone captured images with not much perfect lighting and uniformity in the setting. We addressed this dilemma by grayscaling and down-sizing [1936 × 2592] RGB images to [128 × 128] grayscale images to perform feature extraction easier and faster.

The second dilemma is performing several preprocessing steps using different image filtering and morphological operations to be able to extract their geometrical features. We limited preprocessing to background removal operation using the K-means clustering technique. Instead of ten geometrical features, we applied PCA – which extracts the best-scattered variance-based features from image data and extracted two sets of 50 and 150 feature sets.

The third dilemma is the value of accuracy (80–88%), which leaves room for improvement. For classification, we utilized and personalized three different data mining algorithms of DT, RF and SVM. The first three proposed methods are named "PCA+DT", "PCA+RF", and "PCA+SVM".

We also balanced our feature datasets using Weka Class Balancer and Resample filters and performed feature selection using the arrtibuteSelector feature in Weka. We achieved the accuracy of 100% and 100% for PCA+DT, 100% and 100% for PCA+RF, and 98.20% and 98.50% for PCA+SVM for Brazilian and Mexican species respectively using 150–features dataset as input. The accuracy of all three algorithms considerably outperforms the results of the two reference systems.

The last two proposed methods in this research have the same baseline as the second reference system. We fed our preprocessed data to two different deep convolutional neural networks. In the second reference model, high–resolution raw images are fed to the neural network as input. We used the lower resolution gray–scaled images as the input (to address the need for high–resolution, high–quality images). We utilized and designed two different architectures. First deep neural network is the same as VGG16 network pre–trained with ImageNet (checkpoint) and fine–tuned with our image dataset. The Second deep neural network architecture is a 7–layer CNN with the same structure as VGG network with lower and smaller convolution and fully–connected layers and trained from scratch using our dataset. We achieved the accuracy of 85.77% and 87.50% for VGG16, and 96.93% and 96.26% for 7–layer CNN for Brazilian and Mexican species respectively using 150–feature dataset as input.

All of our five proposed methods outperform the previously developed systems regarding the accuracy. Since the available dataset is small, powerful deep convolutional neural networks and SVM are not performing as good as tree–based algorithms. This problem will be addressed in future research with more acquired images of kissing bugs.

As mention, due to shortage of suffiecient datasets, the first phase of this research which is classification of kissing bugs versus different bugs was not possible to carry out. We plan to gather more Triatomine bug images from from different clinical trials and researches and form a suitable consistent dataset to be able to design a system for that phase. Regarding the second phase of this research, which current project was dedicated to it, more kissing bug images will be gathered and the proposed framework will be tested and configured for larger dataset. Different feature extraction algorithms, specially the ones which preserve spatial information of the images, will be implemented and tested in the future. Finally since it is very important to catch the bug and treat CD right away, designing and implementing an app for CD vector classification is considered to be done after acquiring perfect identification results for our CNN and SVM algorithm.

Bibliography

[1] OHS. Occupational health and safety: Aha statement warns of chagas disease risk, 2018.

[2] Caryn Bern, Sonia Kjos, Michael J. Yabsley, and Susan P. Montgomery. Trypanosoma cruzi and chagas' disease in the united states. *Clinical Microbiology Reviews*, 24(4):655–681, 2011.

[3] Jenny Telleria and Michel Tibayren. *American Trypanosomiasis Chagas Disease: One Hundred Years of Research*. Elsevier, 2018.

[4] Dietmar Steverding. The history of chagas disease. *Parasites & vectors*, 7(1):317, 2014.

[5] CDC. Centers for disease control and prevention: Parasites - american trypanosomiasis (also known as chagas disease), 2020.

[6] World Health Organization (WHO). Chagas disease (American trypanosomiasis), 2019.

[7] A. Moncayo and M. I. Ortiz Yanine. An update on Chagas disease (human American trypanosomiasis). *Annals of Tropical Medicine and Parasitology*, 100(8):663–677, 2006.

[8] Pan Amercian Health Organization. Neglected infectious diseases in the Americas: Success stories and innovation to reach the neediest, 2016.

[9] Chris J. Schofield, Jean Jannin, and Roberto Salvatella. The future of Chagas disease control. *Trends in Parasitology*, 22(12):583–588, 2006.

[10] Jose Rodrigues Coura and Pedro Albajar Vinas. Chagas disease: A New Worldwide Challenge. *Acta Tropica*, 115(1-2):14–21, 2010.

[11] Eric Dumonteil, Maria Elena Bottazzi, Bin Zhan, Michael J. Heffernan, Kathryn Jones, Jesus G. Valenzuela, Shaden Kamhawi, Jaime Ortega, Samuel Ponce De Leon Rosales, Bruce Y. Lee, Kristina M. Bacon, Bernhard Fleischer, B. T. Slingsby, Miguel Betancourt Cravioto, Roberto Tapia-Conyer, and Peter J. Hotez. Accelerating the development of a therapeutic vaccine for human Chagas disease: Rationale and prospects. *Expert Review of Vaccines*, 11(9):1043–1055, 2012.

[12] Maria Carmo Pereira Nunes, Andrea Beaton, Harry Acquatella, Caryn Bern, Ann F Bolger, Luis E Echeverria, Walderez O Dutra, Joaquim Gascon, Carlos A Morillo, Jamary Oliveira-Filho, et al. Chagas cardiomyopathy: an update of current clinical knowledge and management: a scientific statement from the american heart association. *Circulation*, 138(12):e169–e209, 2018.

[13] Qin Liu and Xiao Nong Zhou. Preventing the transmission of American trypanosomiasis and its spread into non-endemic countries. *Infectious Diseases of Poverty*, 4(1):1–11, 2015.

[14] Texas Department of Health Care Services. Chagas Disease, 2019.

[15] Charles B. Beard, Greg Pye, Frank J. Steurer, Ray Rodriguez, Richard Campman, A. Townsend Peterson, Janine Ramsey, Robert A. Wirtz, and Laura E. Robinson. Chagas disease in a domestic transmission cycle in Southern Texas, USA. *Emerging Infectious Diseases*, 9(1):103–105, 2003.

[16] Sonia A. Kjos, Karen F. Snowden, and Jimmy K. Olson. Biogeography and Trypanosoma cruzi infection prevalence of chagas disease vectors in Texas, USA. *Vector-Borne and Zoonotic Diseases*, 9(1):41–49, 2009.

[17] Caryn Bern and Susan P. Montgomery. An Estimate of the Burden of Chagas Disease in the United States. *Clinical Infectious Diseases*, 49(5):e52–e54, 2009.

[18] Rachel Curtis-Robles, Edward J. Wozniak, Lisa D. Auckland, Gabriel L. Hamer, and Sarah A. Hamer. Combining Public Health Education and Disease Ecology Research: Using Citizen Science to Assess Chagas Disease Entomological Risk in Texas. *PLoS Neglected Tropical Diseases*, 9(12):1–12, 2015.

[19] Anis Rassi, Anis Rassi, and José Antonio Marin-Neto. Chagas disease. *The Lancet*, 375(9723):1388–1402, 2010.

[20] Aglaêr A. Nóbrega, Marcio H. Garcia, Erica Tatto, Marcos T. Obara, Elenild Costa, Jeremy Sobel, and Wildo N. Araujo. Oral transmission of chagas disease by consumption of Açaí palm fruit, Brazil. *Emerging Infectious Diseases*, 15(4):653–655, 2009.

[21] Nicole Klein, Ivy Hurwitz, and Ravi Durvasula. Globalization of chagas disease: a growing concern in nonendemic countries. *Epidemiology Research International*, 2012.

[22] Herbert B Tanowitz, Louis V Kirchhoff, Douglas Simon, Stephen A Morris, Louis M Weiss, and Murray Witfner. Chagas ′ Disease. *CLINICAL MICROBIOLOGY REVIEWS*, 5(4):400–419, 1992.

[23] Maria Carmo Pereira Nunes, Wistremundo Dones, Carlos A. Morillo, Juan Justiniano Encina, and Antônio Luiz Ribeiro. Chagas disease: An overview of clinical and epidemiological aspects. *Journal of the American College of Cardiology*, 62(9):767–776, 2013.

[24] Anna M. Afonso, Mark H. Ebell, and Rick L. Tarleton. A Systematic Review of High Quality Diagnostic Tests for Chagas Disease. *PLoS Neglected Tropical Diseases*, 6(11), 2012.

[25] Alejandro G. Schijman, Margarita Bisio, Liliana Orellana, Mariela Sued, Tomás Duffy, Ana M. Mejia Jaramillo, Carolina Cura, Frederic Auter, Vincent Veron, Yvonne Qvarnstrom, Stijn Deborggraeve, Gisely Hijar, Inés Zulantay, Raúl Horacio Lucero, Elsa Velazquez, Tatiana Tellez, Zunilda Sanchez Leon, Lucia Galvão, Debbie Nolder, María Monje Rumi, José E. Levi, Juan D. Ramirez, Pilar Zorrilla, María Flores, Maria I. Jercic, Gladys Crisante, Néstor Añez, Ana M. de Castro, Clara I. Gonzalez, Karla Acosta Viana, Pedro Yachelini, Faustino Torrico, Carlos Robello, Patricio Diosque, Omar Triana Chavez, Christine Aznar, Graciela Russomando, Philippe Büscher, Azzedine Assal, Felipe Guhl, Sergio Sosa Estani, Alexandre DaSilva, Constança Britto, Alejandro Luquetti, and Janis Ladzins. International study to evaluate PCR methods for detection of Trypanosoma cruzi DNA in blood samples from Chagas disease patients. *PLoS Neglected Tropical Diseases*, 5(1), 2011.

[26] Kárita Cláudia Freitas Lidani, Fabiana Antunes Andrade, Lorena Bavia, Flávia Silva Damasceno, Marcia Holsbach Beltrame, Iara J. Messias-Reason, and Thaisa Lucas Sandri. Chagas disease: From discovery to a worldwide health problem. *Journal of Physical Oceanography*, 49(6):1–13, 2019.

[27] Kevin M. Bonney. Chagas disease in the 21st Century: A public health success or an emerging threat? *Parasite*, 21, 2014.

[28] Gabriel A. Schmunis and Jose R. Cruz. Safety of the blood supply in Latin America. *Clinical Microbiology Reviews*, 18(1):12–29, 2005.

[29] José Rodrigues Coura. The main sceneries of chagas disease transmission. The vectors, blood and oral transmissions - A comprehensive review. *Memorias do Instituto Oswaldo Cruz*, 110(3):277–282, 2015.

[30] Faustino Torrico, Cristina Alonso-Vega, Eduardo Suarez, Patricia Rodriguez, Mary Cruz Torrico, Michele Dramaix, Carine Truyens, and Yves Carlier. Maternal Trypanosoma cruzi infection, pregnancy outcome, morbidity, and mortality of congenitally infected and non-infected newborns in Bolivia. *American Journal of Tropical Medicine and Hygiene*, 70(2):201–209, 2004.

[31] P. V. Chin-Hong, B. S. Schwartz, C. Bern, S. P. Montgomery, S. Kontak, B. Kubak, M. I. Morris, M. Nowicki, C. Wright, and M. G. Ison. Screening and treatment of chagas disease in organ transplant recipients in the United States: Recommendations from the chagas in transplant working group. *American Journal of Transplantation*, 11(4):672–680, 2011.

[32] Belkisyolé Alarcón de Noya, Zoraida Díaz-Bello, Cecilia Colmenares, Raiza Ruiz-Guevara, Luciano Mauriello, Reinaldo Zavala-Jaspe, José Antonio Suarez, Teresa Abate, Laura Naranjo, Manuel Paiva, Lavinia Rivas, Julio Castro, Juan Márques, Iván Mendoza, Harry Acquatella, Jaime Torres, and Oscar Noya. Large Urban Outbreak of Orally Acquired Acute Chagas Disease at a School in Caracas, Venezuela. *The Journal of Infectious Diseases*, 201(9):1308–1315, 2010.

[33] David R Moser, LV Kirchhoff, and JE Donelson. Detection of trypanosoma cruzi by dna amplification using the polymerase chain reaction. *Journal of clinical microbiology*, 27(7):1477–1482, 1989.

[34] Nancy R Sturm, Wim Degrave, Carlos Medicis Morel, Larry Simpson, et al. Sensitive detection and schizodeme classification of trypanosoma cruzi cells by amplification of kinetoplast minicircle dna sequences: use in diagnosis of chagas' disease. 1989.

[35] Daniela V Andrade, Kenneth J Gollob, and Walderez O Dutra. Acute chagas disease: new global challenges for an old neglected disease. *PLoS neglected tropical diseases*, 8(7), 2014.

[36] M.V. Cardinal, M.A. Lauricella, P.L. Marcet, and Et al. Impact of community-based vector control on house infestation and Trypanosoma cruzi infection in Triatoma infestans, dogs and cats in the Argentine Chaco. *Acta tropica*, 103(3):201–211, 2007.

[37] Sahotra Sarkar, Stavana E. Strutz, David M. Frank, Chissa Louise Rivaldi, Blake Sissel, and Victor Sánchez-Cordero. Chagas disease risk in Texas. *PLoS Neglected Tropical Diseases*, 4(10), 2010.

[38] Bruce Y. Lee, Kristina M. Bacon, Maria Elena Bottazzi, and Peter J. Hotez. Global economic burden of Chagas disease: A computational simulation model. *The Lancet Infectious Diseases*, 13(4):342–348, 2013.

[39] Jennifer K. Peterson, Sarah M. Bartsch, Bruce Y. Lee, and Andrew P. Dobson. Broad patterns in domestic vector-borne Trypanosoma cruzi transmission dynamics: synanthropic animals and vector control Quantitative analysis of strategies to achieve the 2020 goals for neglected tropical diseases: where are we now? *Parasites and Vectors*, 8(1):1–10, 2015.

[40] Zulma M. Cucunubá, Pierre Nouvellet, Jennifer K. Peterson, Sarah M. Bartsch, Bruce Y. Lee, Andrew P. Dobson, and Maria Gloria Basáñez. Complementary paths to chagas disease elimination: The impact of combining vector control with etiological treatment. *Clinical Infectious Diseases*, 66(Suppl 4):S293–S300, 2018.

[41] Lucia C Orantes, Carlota Monroy, Patricia L Dorn, Lori Stevens, Donna M Rizzo, Leslie Morrissey, John P Hanley, Antonieta Guadalupe Rodas, Bethany Richards, Kimberly F Wallin, et al. Uncovering vector, parasite, blood meal and microbiome patterns from mixed-dna specimens of the chagas disease vector triatoma dimidiata. *PLoS neglected tropical diseases*, 12(10), 2018.

[42] Nathan A Baird, Paul D Etter, Tressa S Atwood, Mark C Currey, Anthony L Shiver, Zachary A Lewis, Eric U Selker, William A Cresko, and Eric A Johnson. Rapid snp discovery and genetic mapping using sequenced rad markers. *PloS one*, 3(10), 2008.

[43] Rodrigo Gurgel-Gonçalves, Ed Komp, Lindsay P. Campbell, Ali Khalighifar, Jarrett Mellenbruch, Vagner José Mendonça, Hannah L. Owens, Keynes de la Cruz Felix, A. Townsend Peterson, and Janine M. Ramsey. Automated identification of insect vectors of Chagas disease in Brazil and Mexico: The Virtual Vector Lab. *PeerJ*, 2017(4):1–23, 2017.

[44] Ali Khalighifar, Ed Komp, Janine M. Ramsey, Rodrigo Gurgel-Gonçalves, and A. Townsend Peterson. Deep Learning Algorithms Improve Automated Identification of Chagas Disease Vectors. *Journal of medical entomology*, 56(5):1404–1410, 2019.

[45] Martin Abadi, Paul Bahram, Ianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, and Et al. TensorFlow: A System for Large-Scale Machine Learning. *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*, 101(C):582–598, 2016.

[46] Diing DM Agany, Jose E Pietri, and Etienne Z Gnimpieba. Assessment of vector-host-pathogen relationships using data mining and machine learning. *Computational and Structural Biotechnology Journal*, 2020.

[47] David Moher, Alessandro Liberati, Jennifer Tetzlaff, Douglas G Altman, Prisma Group, et al. Preferred reporting items for systematic reviews and meta-analyses: the prisma statement. *PLoS med*, 6(7):e1000097, 2009.

[48] Daryl D Cruz, Elizabeth Arellano, Dennis Denis Ávila, and Carlos N Ibarra-Cerdeña. Identifying chagas disease vectors using elliptic fourier descriptors of body contour: a case for the cryptic dimidiata complex. *Parasites & vectors*, 13(1):1–12, 2020.

[49] Hiroyoshi Iwata and Yasuo Ukai. Shape: a computer program package for quantitative evaluation of biological shapes based on elliptic fourier descriptors. *Journal of Heredity*, 93(5):384–385, 2002.

[50] Rodrigo Gurgel-Gonçalves, Ed Komp, Lindsay P. Campbell, Ali Khalighifar, Jarrett Mellenbruch, Vagner José Mendonça, Hannah L. Owens, Keynes de la Cruz Felix, A. Townsend Peterson, and Janine M. Ramsey. Data from: Automated identification of insect vectors of chagas disease in brazil and mexico: the virtual vector lab. 2017.

[51] AR Surabhi, Shwetha T Parekh, K Manikantan, and S Ramachandran. Background removal using k-means clustering as a preprocessing technique for dwt based face recognition. In *2012 International Conference on Communication, Information & Computing Technology (ICCICT)*, pages 1–6. IEEE, 2012.

[52] Kh Tohidul Islam, Ram Gopal Raj, and Ghulam Mujtaba. Recognition of traffic sign based on bag-of-words and artificial neural network. *Symmetry*, 9(8):138, 2017.

[53] Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice-Hall, Inc., 2007.

[54] Pasi Luukka. Classification based on fuzzy robust pca algorithms and similarity classifier. *Expert systems with applications*, 36(4):7463–7468, 2009.

[55] Alaa Eleyan and Hasan Demirel. *Pca and lda based neural networks for human face recognition*, volume 558. INTECH Open Access Publisher, 2007.

[56] Tinku Acharya and Ajoy K Ray. *Image processing: principles and applications*. John Wiley & Sons, 2005.

[57] Ronaldo C Prati, Gustavo EAPA Batista, and Maria Carolina Monard. Data mining with imbalanced class distributions: concepts and methods. In *IICAI*, pages 359–376, 2009.

[58] Eibe Frank. Oversampling and undersampling. `https://waikato.github.io/weka-blog/posts/2019-01-30-sampling/`, 2019.

[59] Thales Sehn Korting. C4. 5 algorithm and multivariate decision trees. *Image Processing Division, National Institute for Space Research–INPE Sao Jose dos Campos–SP, Brazil*, 2006.

[60] Neeraj Bhargava, Girja Sharma, Ritu Bhargava, and Manish Mathuria. Decision tree analysis on j48 algorithm for data mining. *Proceedings of International Journal of Advanced Research in Computer Science and Software Engineering*, 3(6), 2013.

[61] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[62] Peijun Du, Alim Samat, Björn Waske, Sicong Liu, and Zhenhong Li. Random forest and rotation forest for fully polarized sar image classification using polarimetric and spatial features. *ISPRS Journal of Photogrammetry and Remote Sensing*, 105:38–53, 2015.

[63] Alexander Verner. Lstm networks for detection and classification of anomalies in raw sensor data. 2019.

[64] V Rodriguez-Galiano, M Sanchez-Castillo, M Chica-Olmo, and MJOGR Chica-Rivas. Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines. *Ore Geology Reviews*, 71:804–818, 2015.

[65] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[66] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[67] Faisal Khan, Frieder Enzmann, and Michael Kersten. Multi-phase classification by a least-squares support vector machine approach in tomography images of geological samples. 2016.

[68] Allison Y Hsiang, Anieke Brombacher, Marina C Rillo, Maryline J Mleneck-Vautravers, Stephen Conn, Sian Lordsmith, Anna Jentzen, Michael J Henehan, Brett Metcalfe, Isabel S Fenton, et al. Endless forams:> 34,000 modern planktonic foraminiferal images for taxonomic training and automated species recognition using convolutional neural networks. *Paleoceanography and Paleoclimatology*, 34(7):1157–1177, 2019.

[69] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[70] Lars Hertel, Erhardt Barth, Thomas Käster, and Thomas Martinetz. Deep convolutional neural networks as generic feature extractors. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–4. IEEE, 2015.

[71] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[72] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[73] Jiawei Han, Micheline Kamber, and Jian Pei. Data mining concepts and techniques third edition. *Morgan Kaufmann*, 2011.