January 2016

Avian Retinal Photoreceptor Detection and Classification using Convolutional Neural Networks

Edward Durham Collier
Purdue University

Graduate School Form
30 Updated 12/26/2015

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Edward Collier

Entitled
Avian Retinal Photoreceptor Detection and Classification using Convolutional Neural Networks

For the degree of Master of Science

Is approved by the final examining committee:

Bedrich Benes
Chair

Esteban Fernandez-Juricic

Tim McGraw

Innfarn Yoo

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy of Integrity in Research" and the use of copyright material.

Approved by Major Professor(s): Bedrich Benes

Approved by: Patrick Connolly 4/15/2016
Head of the Departmental Graduate Program Date

AVIAN RETINAL PHOTORECEPTOR DETECTION AND CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS

A Thesis

Submitted to the Faculty

of

Purdue University

by

Edward D. Collier

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

May 2016

Purdue University

West Lafayette, Indiana

ACKNOWLEDGMENTS

I would like to thank Dr.Ilias Bilionis for his assistance. Without him I would not have understood CNN's or Caffe. Thanks to Kelly Lennginton for taking time to train me to detect and classify retina regions in Image-J. I would like the thank Dr.Esteban Fernandez-Juricic for funding me for a portion of this project and being on my committee. I want to thank my committee members Dr.Innfarn Yoo and Dr.Tim McGraw. I would like to thank Dr.Bedrich Benes. Without his guidance and support I wouldn't have been able to make it as far as I did.

Lastly, I want to thank all my friends and family for supporting me. My friends in the HPCG lab have always been great sources of support and guidance. My mom and siblings for always having my back along this journey. Most importantly, my girlfriend Denise. She has always pushed me and inspired me to do my best. Without her I could never have accomplished all I have.

Special thanks to NVIDIA for providing free hardware and for the GPU Research Center.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABBREVIATIONS

CHT	Circular Hough Transformation
CNN	Convolutional Neural Network
R-CNN	Region-based Convolutional Neural Network
SVM	Support Vector Machine
ROI	Region of Interest
NMS	Non-Maxima Suppression

GLOSSARY

loss function	a function that calculates the inherent error of classification as a result of noise or other factors.
max pooling	partitions the input image into rectangular sub-images and finds the maximum value of each.

ABSTRACT

Collier, Edward D. MS, Purdue University, May 2016. Avian Retinal Photoreceptor Detection and Classification using Convolutional Neural Networks. Major Professor: Bedrich Benes.

Avian retinas contain special light filtering cones called photoreceptors. These photoreceptors help filter out specific wavelengths of light, giving birds a good range of distinction between colors. There are five distinct types of photoreceptors: red, yellow, transparent, colorless and principle. A specific photoreceptor can be identified by an organelle called an oil droplet. Dectecting and classifying the oil droplets is currently done by hand which can be a time consuming process. Using computer vision detecting and classifying the photoreceptors can be done automatically. The recent introduction of deep learning in computer vision has revolutionized automatic classification, producing classification results identical to what a human could do. Using deep learning the human element can be eliminated from oil droplet detection and classification. It can take days for a human to process and entire retina, but using deep learning a computer can perform the same take in a matter of minutes.

In this work, using current state of the art deep learning frameworks we have created a Convolutional Neural Network that can classify the photoreceptors in the microscope images identical to human classified images. Coupling the CNN with various object detection algorithms, including R-CNN and two stage Hough transform, potential oil droplets can be both detected and classified automatically. Our results show that a CNN trained to classify the five different has human level accuracy. Detection algorithms still lag behind classification in accuracy with the best algorithms obtaining obtaining only 66% on the PASCAL VOC 2012 data-set.

Despite the limitations in detection, we show that adjusting the color contrast of the retina images accurate detection can be achieved for oil droplets regions.

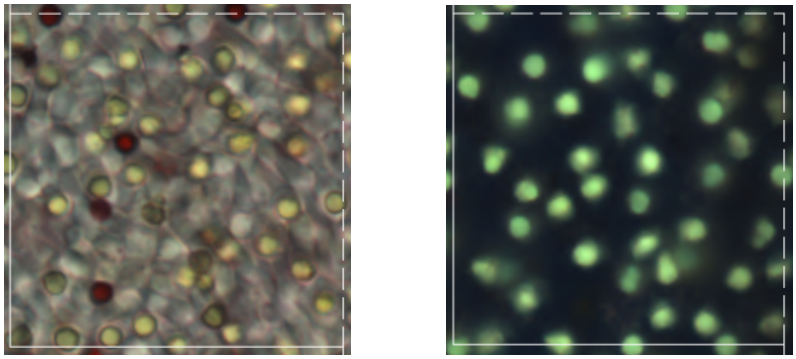
CHAPTER 1. INTRODUCTION

The eye is one of the most complex body parts in all of anatomy. Computer Graphics revolves around how the human eye perceives input, and how it can be deceived. However complex the human eye might be, the eyes of birds are equally as complex. Arguably one of the most important body parts on a bird, the birds ability to see is essential to its ability to fly. Avian retinas have adapted over time to accommodate for flight. Unlike humans, who can only perceive color between 360 nm and 720 nm in the spectral band, birds can perceive a wider range of color that can extend down to ultraviolet light in some birds (Bowmaker, 1980).

One of the most important and distinctive features of avian retinas is for the photoreceptors in the retina to selectively filter out light. These filters are commonly known as oil droplets (Vorobyev, 2003). In an avian retina there is one single type rod photoreceptor, four single type cone, one spectral type double cone, and double cones made up of a principle and secondary component (Hart & Hunt, 2007). Of these only the single cones and the double cones principle component contain oil droplets (Hart & Hunt, 2007). The four types of single cone oil droplets are Red (R-Type), Yellow (Y-Type), Colorless (C-Type), and Transparent (T-Type). R-Type oil droplets are contained in photoreceptors that absorb long wavelengths(Fernández-Juricic et al., 2013). Photoreceptors sensitive to medium wavelengths contain Y-Type oil droplets (Fernández-Juricic et al., 2013). C-Type oil droplets are contained in photoreceptors that absorb short wavelengths (Fernández-Juricic et al., 2013). T-Type oil droplets absorb light in the ultraviolet range (Fernández-Juricic et al., 2013). The double cone principle component (P-Type) is also sensitive to long wavelengths (Fernández-Juricic et al., 2013).

The goal of this thesis is to develop a multi-method system for detection, and a CNN for automatic classification of oil droplets. Oil droplets are observable

features that are associated with a specific photoreceptor. Biologists use microscope images of regions of avian retinas to observe oil droplets because the photoreceptors themselves can not be clearly seen (Hart, 2001b). Each microscope image of a retina region contains two images, a bright field and an epi image. The bright field image is a colored picture of the cell region that shows the oil droplet colors. The epi image is the same region just under a florescent light; this shows cells that cant be identified or detected by just color alone. Figure 1.1 shows what a bright field and epi pair might look like for a retina region.



(a) Bright field (b) Epi

Figure 1.1. Paired bright field and epi images

At Purdue University, the Biological Sciences Department has an entire research team that is studying bird vision. The department has hundreds of retinas that need to be processed. This includes detecting, classifying and counting the cells in a microscope image of the retina. This is a very time consuming process. Computer vision can be used to speed up research by helping remove the human element from the detecting, categorizing and counting. We hypothesize that, using a convolutional neural network (CNN) combined with other techniques from computer vision, a computer can perform the same task in a matter of seconds for an entire eye. If successful, this new method would useful to biologist by significantly reduce the amount of time it takes to count oil droplets.

1.1 Problem Statement

The classification of oil droplets in avian retinas is a crucial task for research in avian vision (Fernández-Juricic et al., 2013). To observe oil droplets researchers must dissect an avian retina and take photos of the retina under a microscope. While the images vary in quality it is still critical that the oil droplets in all the undamaged retina images be classified. Currently this task is done manually by trained researchers. Training a researcher to correctly classify images takes time, however. Additionally, once a researcher has been trained it can still take days to classify all the images for one retina. Automating the classification of avian oil droplets with a computer would offer a significant speed up for avian vision research. Not only would this eliminate the time required to train a researcher, but a computer could detect and classify the images in a fraction of the time.

1.2 Scope

The scope of this thesis covers two main portions; detecting and categorizing oil droplets in avian retinas. This means that the method being created limits itself to detecting cells present in an image. After detection, the application will identify the cells in the image as one of the oil droplet types.

This project creates two convolutional neural networks (CNNs). The first network is created on images of a single oil droplet. Its purpose is to show that a CNN trained on enough data can perform human levels of classification. The second network is a fast R-CNN. This network is built to detect and classify cells in the full image. The single cell network gives a best case classification scenario where the fast R-CNN gives the realistic results for detection and classification for the full image.

While CNNs are the sole method used for classification there are three detection methods implemented. The first is the R-CNN which has already been described. Second is k-means segmentation with boundary tracing on an adjusted

image. The last detection method implemented is a two stage Hough transformation on an adjusted image.

All images tested were processed independent from all other images in the retina. This means that regardless of what the results from other processed images is, they will have no effect on how another image is processed. The images used for training the network are not independent. In CNN training the results of one training image will affect the weights of the network which will have an effect on future results for better or worse.

The research and specific scientific details of bird retinas are not in the scope of this thesis. The specific uses of each type of cell is irrelevant to categorization and detection, and thus is not considered. The algorithms presented in this thesis are designed to show the effectiveness of CNNs for the task presented. Due to the amount of training data used for each CNN this paper does not claim them to be the optimal CNNs. With significantly greater training data the network error would significantly decrease.

1.3 Significance

Detecting, classifying and counting oil droplets in avian retinas is currently done by hand. At Purdue University researchers use a tool called ImageJ to manually label images (Abràmoff, Magalhães, & Ram, 2004). Using ImageJ detection, categorization and counting is done by hand and is very time consuming. Processing all the images for one retina might take days or weeks, and Purdue has hundreds of these retinas that need processing. Automating detection and classification could greatly speed up research at Purdue and beyond, reducing processing time to a matter of seconds and not days.

Deep Learning has proven to be a useful method for cancer cell detection in the medical field (Cruz-Roa, Ovalle, Madabhushi, & Osorio, 2013). Deep Learning has yet to be applied to avian retinas to detect and classify oil droplets. This

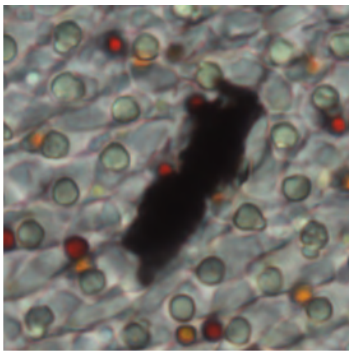
project has given the researcher the opportunity to apply Deep Learning in a new way and expand the knowledge and application base of Deep Learning and other computer vision techniques.

1.4 Research Question

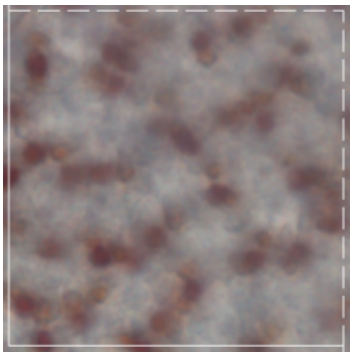
Using the most current image processing techniques in computer vision, can a Convolutional Neural Network be trained that can detect and categorize the cells in a microscope image of an avian retina, with 90% accuracy?

1.5 Assumptions

- Image must contain all five oil droplets.
- Images must be absent of any rips or tears. Faulty images can occur, but this application will not account for them and it is left to the user to filter those out. Figure 1.2(a) gives an example of a damaged retina region that would be unusable.
- Images must have a high enough resolution to visual identify cells in the image. Figure 1.2(b) gives and example of an image with poor resolution.
- Every bright field image has an epi image.
- All the images used as input are avian retinal images, no other animal is used in this study.
- All cells in the image fall into either red, yellow, principal, colorless, or transparent.
- All libraries and tools used to build this application work correctly.



(a) Retina region damaged during dissection



(b) Retina region that is to blurry to distinguish cell types.

Figure 1.2. Sample images of retina regions that can not be processed

1.6 Limitations

- The application can only classify oil droplets as either red, yellow, transparent, colorless or principle.
- CNN accuracy is dependent human input. If any of the training images are formatted incorrectly, such as mislabeled, then the accuracy of the application will suffer.
- The amount of training that can be performed is directly dependent on the number of dissections the Biological Sciences department has performed.
- The application is specifically built for the images provided by the Purdue Biological Sciences department. All input images must mimic the types of images the Biological Science department produces.
- The accuracy of the CNN reflects the amount and quality of the training data used. Most CNNs require tens of thousands of training and testing images to converge. For the single cell network approximately 6,000 images were processed and used. For the fast R-CNN network only approximately 500

regions of interest (ROI) were used for training. The lack of training data is primarily a result of time and man power constraints.

- This thesis relies heavily on several algorithms like selective search (Uijlings, van de Sande, Gevers, & Smeulders, 2013). While these algorithms are assumed to work as intended it cannot be assumed they are perfect solutions. The algorithms used represent the best available methods for performing the tasks outlines in this thesis.
- This study is not concerned with bird species.
- Animals other than birds are not considered.
- This study is not concerned with the use of the information the application provides, but the application itself.
- This study is concerned with the effectiveness of CNNs for detecting and classifying avian retina oil droplets. It does not evaluate the user ability of the application created.

1.7 Summary

This chapter has provided information on why this thesis is important and its background. The scope of the project has been outlined along with relevant definitions and the limitations this thesis faces. The next chapter will review previous research in the fields of computer vision, machine learning and cell biology.

CHAPTER 2. REVIEW OF RELEVANT LITERATURE

This chapter provides an overview of literature across the fields of avian retinal research in biology and computer vision. At Purdue University there are researchers who are taking microscope images of avian retinal regions and need to detect, classify and count the visual pigment cells in them. This process is currently done by hand using ImageJ. Using ImageJ involves manually aligning two images of the retina region, one regular image and one fluorescent, and then switching between the two images to determine cell classification. This is a long and time consuming process that slows down research considerably. Additionally, it takes a considerable amount of time to train a human to see the differences in the cell types. Instead of training a human, computer vision and convolutional neural networks (CNN) can be used to produce similar results in a fraction of the time.

This research applies existing tools and algorithms in computer vision and machine learning to process the microscope images. While these tools are combined and applied in new ways, the goal is not to recreate these tools. Caffe deep learning framework created by the Berkley Vision and Learning Center was used to create the deep learning model used in the application (Jia et al., 2014). Fast R-CNN (R. Girshick, 2015), which is a CNN that is specially designed for detection problems, was used to perform both detection and classification. Many of the computer vision algorithms used were implemented by Matlab's computer vision toolbox.

2.1 Research on Bird Retinas

Using microscopic images, avian visual pigment can be examined in several different ways. Nathan S. Hart provides an good overview of the five oil droplets,

> "Birds have five different types of visual pigment and seven different types of photoreceptors-rods, double (uneven twin) cones and four types of single cone." (Hart, 2001b)

This information is primary to the research being conducted in this paper. The five types of visual pigment are the cells the application will be detecting and classifying. With this information (Hart, 2001b) measures various aspects of spectral sensitivity including wavelengths of pigment. This represents an example of some of the research the the applications created in this thesis will help with.

The single cone photoreceptors in an avian retina are the only cones that handle color distinction, where the double cones handle tasks such as movement detection (Hart & Hunt, 2007). The evolution of the visual pigments is observed by (Hart & Hunt, 2007), finding similarities to visual pigment genes in other vertebrates. They conclude these similarities go back to a common ancestor in early vertebrate evolution. Oil droplets filter out specific wavelengths of light allowing a bird to better distinguish colors (Vorobyev, 2003). Because of their benefits to color vision oil droplets have been retained through evolution (Vorobyev, 2003) .

Avian photoreceptors and visual pigment can be examined in three aspects (Hart, 2001a), but this thesis focuses on their first method, the measurements of single cells in the images. While this project is deeply connected with the background research in avian visual pigment and photoreceptors, the results of such research is not the concern. This thesis is concerned with developing a tool for avian retinal researchers, the development of algorithms for cell detection, and the development of optimization algorithms for cell identification. These goals require a knowledge of the information contained in the microscope images, but is not concerned with the intended use of such information. The researcher directs the reader to (Hart, 2001a) for more information regarding research into avian photoreceptions and visual pigment from a biological sciences perspective.

2.2 Biological Image Segmentation

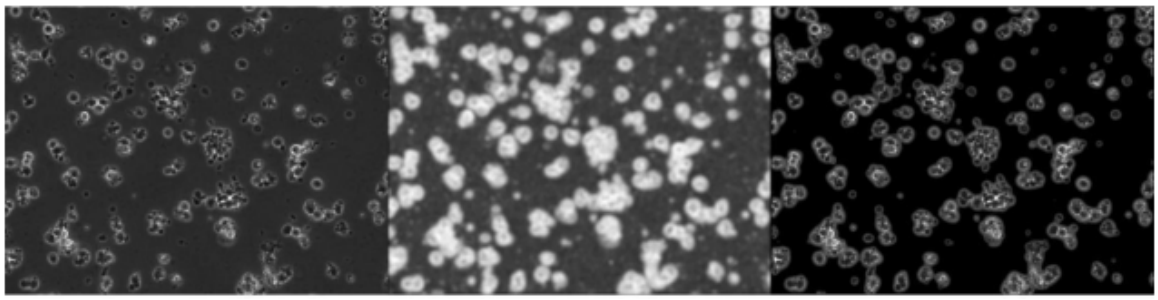


Figure 2.1. Cell detection using foreground/background segmentation from (Guan et al., 2014).

Using computer vision to detect cells in microscope images is used by (Guan et al., 2014). The authors seek to take microscope images of human embryo stem cell regions and detect cell regions. The purpose of the paper by (Guan et al., 2014) is to develop an improved automatic algorithm to complete the detection. This goal aligns itself with the goal of this thesis, however the researchers goal is to detect the cells in avian retinas.

Image gradient has been used to distinguish cells from the background of the image by (Guan et al., 2014). Gaussians are used to identify likely cells using Grey Level. Noise is filtered using a Gaussian mask. With a list of likely cells and the distinction between foreground and background as metrics the system decides what is and is not a cell, shown in figure 2.1. The distinction between foreground and background using gradient is important. Using the method detailed by (Guan et al., 2014) the possibility of false positive cells during both detection and classification can be eliminated.

Important to this thesis is the ability to detect partial circles. In many case the cells are not perfect circles in the images. For this reason methods must be looked at to detect partial circles. Using active contours to detect the iris in biometric scans, (Ross & Shah, 2006) are able to identify partial circles in images.

Following an initial filter for circle detection the candidate circles whose diameter is larger than a certain threshold in the image is discarded.

2.3 Histograms

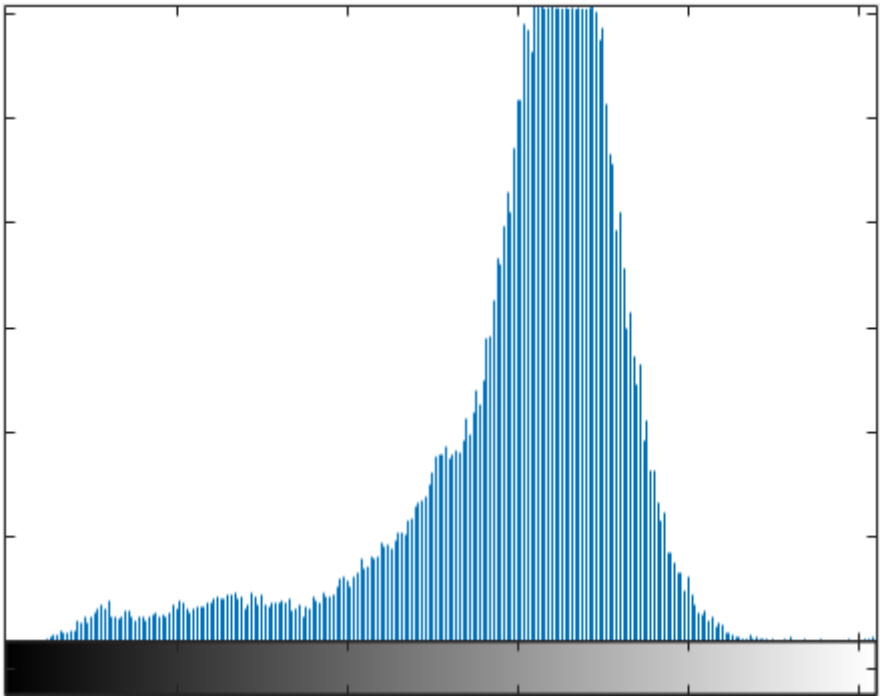


Figure 2.2. Grey scale image histogram of a retina cell region. Columns represent the number of pixels of a certain value.

Image histograms can be used to plot the distribution of pixels of different tones in an image or region of an image. For both the epi and bright field images of a retina a histogram can be used to distinguish cells from the background. This can be especially effective in the epi images that only use two colors; one for the cells, and another for the background.

Histograms can go beyond differentiating the foreground from the background. An image histogram can be used to aid in classification. In a bright field image one can distinguish red and yellow visual pigment cells from other visual pigment cells based off the color of the cell. Similar techniques were used to detect

and classify smoke particles by (Pahalawatta & Green, 2012). Using a three channeled image histogram with an R, G, and B channel the pixel colors of a detected cell can be observes. If the histogram for a specific cell contains primarily R pixels and little of the others then its initial classification can be red, if a cell has high G and R then its initial classification can be yellow. R-Type and Y-Type are the only cells that are classified by color and thus the only ones that can be classified specifically by an image histogram.

A specific type of histogram known as the radial histogram has important uses in circle detection. Radial histograms compute the number of dark pixels about the edge of an object around its center (Kavallieratou, Sgarbas, Fakotakis, & Kokkinakis, 2003). Radial Histograms have lent themselves to a variety of image segmentation applications. Radial histogram are used by (Bertozzi & Broggi, 1997) in their self driving car implementation, while (Tang, 2011) used radial histograms to detect hands and extended fingers on the Microsoft Kinect. Radial histograms have be shown to perform well for detection of symmetric objects (Davies, 2004). This makes them an important tool for circle detection when used with a Hough Transformation.

2.4 Hough Transformation

Hough transformation, created and patented by (Hough, 1962), is used for pattern detection in pictures. Hough transform excels in shape detection. Detecting lines, curves and circles is easy using Hough transform (Tian, Pan, Cheng, & Gao, 2004). The basic idea of Hough transform is that the algorithm transforms an image from image space to parameter space. The algorithm then makes a decision on whether certain points make a shape or not. The technical aspects of Hough transform are not a focus of this thesis, however, we direct the reader to (Tian et al., 2004) and (Hough, 1962).

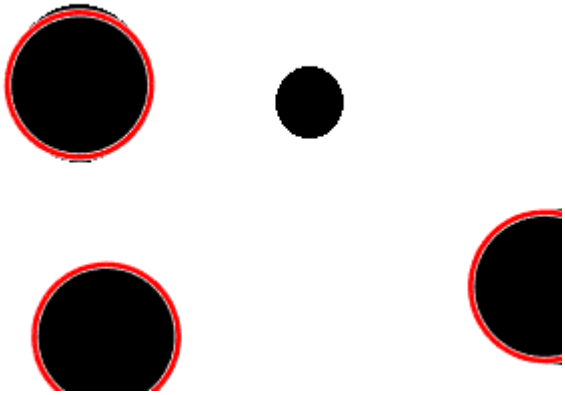


Figure 2.3. Circle detection using Hough transformation. Even partial circles can be detected.

Hough transform is a good method for circle detection (Duda & Hart, 1972). While the algorithm is relatively old, it is still used in current research for feature detection. Recent work in retinal scanning use Hough transform for circle detections to identify the iris of a human eye (Bakshi, Mehrotra, & Majhi, 2011).

The function imfindcircles in Matlab is an extensive method for circle detection in images that uses a Circular Hough Transformation (CHT). This function implements two circle detection algorithms, phase coding and two stage. Since this function works specifically for images the 3-D accumulator array that holds candidate radii and votes for each in the standard 3-D Hough Transformation is only 2-D for both methods, this provides a significant speed up. Additionally, both methods explicitly use edge pixels and their gradients to help make determinations. Phase coding, initially described by (Atherton & Kerbyson, 1999), encodes the estimated radius as phases information in the accumulator array. Decoding phase information from the proposed circle center can then estimate the radius (Davies, 2004). The two stage method creates a radial histogram that computes the radius of a circle using proposed center and edge information (Yuen, Princen, Illingworth, & Kittler, 1990).

2.5 K-Means Clustering

K-means clustering is a clustering algorithm that takes n observations and clusters them based off some mean into K clusters. Using image histograms k-means clustering can be done automatically by clustering the pixel values. When taking a histogram of a region of an image the peak, or mean of the colors can be used to cluster (Chen, Chen, & Chien, 2008). Using several iterations of k-means the background can be filtered out of the retinal cell regions leaving the cells filtered out for detection.

K means is commonly used for segmentation for retinal images and scans. Using multi-space cluster, which includes k-means, (Ram & Sivaswamy, 2009) detects Exudates in optical fundus images. Automatic retinal vessel segmentation using k means is performed by (Lupaşcu & Tegolo, 2010), by clustering a self-organizing map into two classes.

2.6 SIFT

SIFT is a computer vision algorithm for object matching that is invariant to transformations (Lowe, 1999). SIFT is a trained algorithm, that matches objects of interest in an input image to an initial training image, or images, with the same detected objects (Lowe, 1999). Color clarity is less important for SIFT because SIFT works best with gray (colorless) images (Y. Zhang, Zhaoxing, & Han, 2009).

A method for computing the location of objects in images using SIFT was proposed by (Sivic, Russell, Efros, Zisserman, & Freeman, 2005). This method works well as an object proposal method for images, but runs the risk of detecting the same object multiple times in noisy images. Using a combination of graph cuts and SIFT objects in images can be segmented and recognized (Suga, Fukuda, Takiguchi, & Ariki, 2008). This is a novel approach when objects are distinct and vary from the background. For this thesis SIFT was shown to have trouble

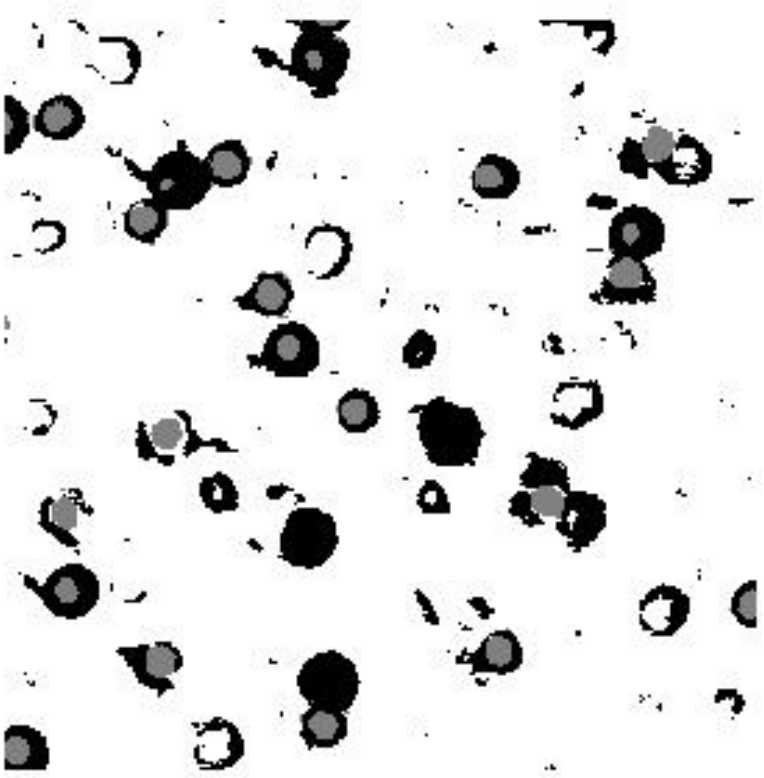


Figure 2.4. oil droplets segmented from the background, drawn white, using k-means image segmentation.

extracting features from the cells in the images, especially for some of the lower quality images.

2.7 Decision Tree Learning

Decision tree learning is one of the most widely used and well known machine learning methods for classification. Decision tree learning uses a tree to

map observations to a specific result. This can be an incredibly powerful tool for classification, even in biology.

Decision tree learning is used to classify ovarian cells as cancerous or not in (Vlahou, Schorge, Gregory, & Coleman, 2003). Using trained and known observations the researchers created a decision tree that would identify a cell as cancerous or not. The decision trees used by the researchers only classify between non-cancer and cancerous cells. Different from (Vlahou et al., 2003) cells in this thesis must be classified into one of five different types. Decision tree learning can classify into multiple types but is also effective in binary classification.

2.8 Cell Distribution

When detecting or classifying cells distribution can be a key variable. One of the more common ways to describe cell distribution is through the Poisson distribution. Poisson Distributions were used by (Knudson, 1971) to predict the number of retinoblastomas an individual will produce in the event of a specific mutation. The distribution of cells and their probability of occurrence is relative to its neighbors is described by (Rodieck, 2003). For more information on the Poisson Distribution or other methods for describing distribution (Rohatgi, 2003) provides a good overview.

2.9 Convolutional Neural Networks

Convelutional neural networks first gained use for classification in the 1980's and 1990's (LeCun et al., 1989, 1998). However, the CNNs used then were not nearly as effective as the ones used today. In fact during the mid 2000's support vector machines (SVM) were the prominent method for image classification. It wasn't until (Krizhevsky et al., 2012) that CNNs became relevant again, and from there have taken off.

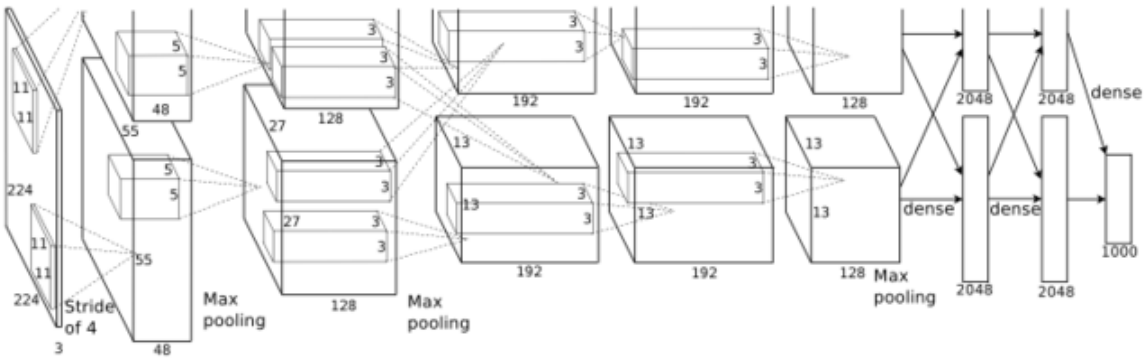


Figure 2.5. Visualization of the AlexNet architecture from (Krizhevsky et al., 2012)

Research by (Hubel & Wiesel, 1968) found that eyes contain complex blankets of cells. These cells are very sensitive to regions of the visual field and together form filters that are very adapt at pattern recognition and feature detection. This system of filters in animal eyes was the primary influence behind the design of convolutional neural networks. Additionally, the alternating convolutional and sub-sampling architecture in CNNs shown in Figure 3.3 is an idea also borrowed from animal eyes (Duda, Hart, & Stork, 2012; Fukushima & Miyake, 1982). CNNs represent one of the most widely used examples of concepts in biology translating to concepts of computer science and machine learning (Alon, 2003).

CNNs have been found to have a wide range of uses outside of standard object classification. Two of the more classic examples of CNNs being used outside of detection are scene recognition (Zhou, Lapedriza, Xiao, Torralba, & Oliva, 2014) and image style recognition (Karayev et al., 2013). While these two topics are still subsets of classification, they do show the versatility and power of neural networks. Some other uses include age and gender recognition (Levi & Hassner, 2015a) and even emotion recognition (Levi & Hassner, 2015b).

A CNN is a specific type of neural network. There are many types of networks in existence. Each type of network needs to be optimized for different

tasks. Examples of other neural networks include Recurrent networks (RNN) (Donahue et al., 2014) which performs well for videos, and fully convolutional networks (Long, Shelhamer, & Darrell, 2014) which are much deeper than normal CNNs and good for pixel wise predictions. CNNs work well for object recognition in images and can be easily trained with modern GPU's (Krizhevsky et al., 2012). CNNs to date have produced the best results on classification data sets like ImageNet, MNIST and CIFAR (Krizhevsky et al., 2012).

One of the large and more open problems in CNNs is that, while very good at single label images, they are not as proficient with multi-label images. While there are many CNNs that can classify multi-label images they generally require ground truth images with bounding boxes for training. Obtaining large quantities of multi-labeled bounding box images is incredibly time consuming. Hypotheses-CNN-Pooling (HCP), created by (Wei et al., 2014), is a multi-label CNN that does not require bounding box images. The CNN that HCP uses can be trained on single image data set saving valuable time. While HCP can output multiple labels for an image, it is still a classification CNN. For the case of the retinal cell regions it is known that there will be the five types of oil droplets in the image, but their location is what must be found.

Deep learning and CNNs have been particularly useful at classifying cancerous cells. Using deep learning (Cruz-Roa et al., 2013) detect cancer cells from non-cancer cells in microscope images of skin to identify carcinoma. CNNs have been used to identify breast cancer in (Abdel-Zaher & Eldeib, 2016; Rouhi, Jafari, Kasaei, & Keshavarzian, 2015; Wu et al., 1995). While all the cells that will be classified by the CNNs created in this thesis will be normal, the research on cancer cells is important because it shows that neural networks can be trained on microscopic biological data.

2.9.1 CNN Detection

While the initial use of CNNs was for image classification, CNNs are now being put to use for recognition as well. There are two notions of recognition with CNNs, detection and segmentation (Hariharan, Arbeláez, Girshick, & Malik, 2014). Detection finds all the instances of objects in the image and places bounding boxes around the object. Segmentation uses pixel-wise labeling to find all the pixels of a certain object in the image. The algorithms in this thesis will focus strictly on detection, because segmentation offers no notion of the number of instances of an object, just segments of the image that contain pixels associated with that object (Hariharan et al., 2014). While this thesis deals strictly with detection the work of (Xu, Schwing, & Urtasun, 2014) provides good insight into image segmentation and how it is used.

Detection is a very open problem in computer vision. The human eye has adapted over thousands of years to recognize patterns, making it very adept at detecting objects in the field of view. Even with the introduction of deep learning, algorithms for object detection are still not as effective as classification. In fact while CNNs can achieve very high accuracy on the ImageNet classification challenge, top CNNs like GoogLeNet could only achieve 40% to 43.9% accuracy for the ImageNet detection challenge (Szegedy et al., 2014).

For detection with CNNs classification is used to lead to detection. There are two methods for using CNNs for object detection. The first method uses sliding windows to classify sub-images (Sermanet et al., 2013). This method is straight forward, but helps solve a fundamental problem with CNN detection: how can multiple objects in a single image be detected when the CNN is trained for binary classification? By breaking the image into smaller sub-images, each containing presumably one object or the background, the CNN can classify the one object in the sub-image. Sliding window is a straight forward method that can be used on basic CNNs at the cost of speed and efficiency.

The second method for detection is to use Region-based Convolutional Nerual Networks (R-CNN) (R. B. Girshick, Donahue, Darrell, & Malik, 2013). R-CNN extracts feature vectors from regions of interest (ROI) to detect objects using a CNN (R. Girshick, Donahue, Darrell, & Malik, 2014). A number of algorithms can be used to compute ROIs including selective search (Uijlings et al., 2013) or edge boxes (Zitnick & Dollár, 2014). Using the regional proposals a feature vector is created. The feature vector is then classified using SVM's (R. B. Girshick et al., 2013). Originally training R-CNN was slower than training a standard CNN because there was no shared computation for each forward pass of the network (R. Girshick, 2015), but later implementations starting with fast R-CNN have included shared computation. R-CNNs have performed well on the detection challenges like PASCAL VOC 2012, where it achieved 66% accuracy (R. B. Girshick et al., 2013). The main drawback to R-CNN is network training. Not only is the training slower, but the training and testing data must be formatted differently. Whole images must be classified either pixel wise or with bounding box annotations. This is much more time consuming than standard image labeling where an entire image has one label.

A faster and more accurate R-CNN called fast R-CNN is used in this thesis (R. Girshick, 2015). Recently a new R-CNN was released called faster R-CNN (Shaoqing Ren, 2015). Faster R-CNN was released during the development of the algorithms described in this thesis and is thus not considered, but offers an interesting option for the future. See Figure 2.6 for an example of fast R-CNN output.

On the ImageNet 2013 detection challenge fast R-CNN performed the best with 31% while the sliding window algorithm Overfeat (2) came in second with 24% (R. B. Girshick et al., 2013). These results highlight one of the main points about detection algorithms: compared to classification detection is still far behind. Even the best detection algorithms still produce below 50% accuracy results (Gupta, Girshick, Arbeláez, & Malik, 2014).

Since the release of R-CNN there has been vast amounts of research looking to improve detection using R-CNN. Bottom up regional proposals have been used to avoid the need for pre-computed bounding box annotations in R-CNN, while creating a CNN that can detect both an entire object and its parts (N. Zhang, Donahue, Girshick, & Darrell, 2014). Areas of image segmentation have seen improvements by creating fully convolutional networks (FCN) (Long, Shelhamer, & Darrell, 2015). This network replaces the fully connected layers at the end of the standard CNN architecture with more convolutional layers, hence the name fully convolutional.

Figure 2.6. An example or Fast R-CNN detecting cars in an image from (R. Girshick, 2015).

2.10 Conclusion

Avian retinal research is a large field with many applications (Hart, 2001b), but a lack of automated software solutions can slow down the research considerably. Cell detection has been attempted in research before (Guan et al., 2014). Using similar methodology to existing research along with deep learning (R. Girshick et al., 2014) this thesis shows that CNNs can be used to automate the time consuming task of detecting and classifying. This application is flexible and can be extended to a wide range of research projects in biology, not just the ones being conducted at Purdue University.

2.11 Summary

This chapter gives background information on the research being done on avian retinas, where this application fits into that research and the tools this application will employ. A detailed look into how detection and classification will be performed automatically along with an overview on CNNs. Methods for automatic detection using CNN were discussed along with their drawbacks. Additionally, cell localization with Poisson distribution, k-means clustering, image histograms and the Hough transformation are discussed.

CHAPTER 3. FRAMEWORK AND METHODOLOGY

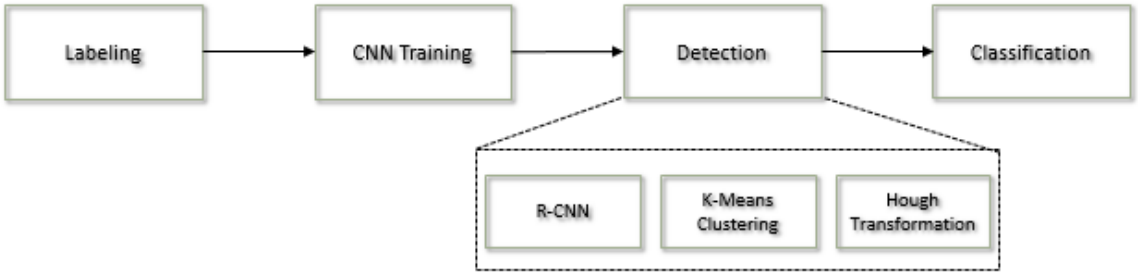


Figure 3.1. An overview of the system implemented in this thesis.

This thesis is a quantitative research project. Figure 3.1 presents the pipeline of our system. The first phase is data labeling. All data used for training the convelutional neural networks created must be specially labeled for input. The second phase trains a CNN using the labeled data produced in the first phase. The detection phase has multiple independent parts. Three detection algorithms, R-CNN, image adjusted k-means segmentation and image adjusted Hough transformation, were implemented in our system. Each of the detection algorithms is independent of the other. Regardless of the detection algorithm used classification is always done by the trained CNN. The oil droplet regions output from detection are all passed through the CNN and classified into either red, yellow, transparent, colorless or principle. If the CNN is an R-CNN then a background class is also present. The effectiveness of the application created is judged by its accuracy in cell classification and detection compared to pre-classified images done by humans.

3.1 Initial Non-Machine Learning Approach

The initial approach to detecting and classifying cells was to use conventional methods in computer vision. These were non-machine learning methods like k-means segmentation and Hough transformations (Hough, 1962; Tian et al., 2004). This approach first sought to detect the cells and the classify all the detections. The epi image in this method was relied on heavily to help with classification and detection. Further iterations of this application used basic machine learning techniques like decision trees to build classifiers for the oil droplets.

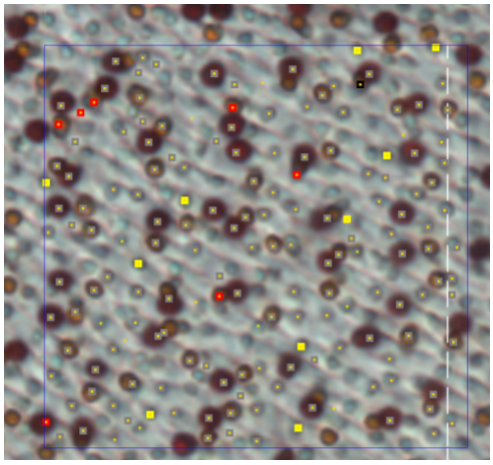


Figure 3.2. Results for initial approach. Notice a majority of the classifications are wrong along with the existence of multiple false positives.

Detection results for the initial approach produced countless false positives and numerous misclassifications. While the detection coverage was good, heavy background noise in the images led to large amounts of false positives. The classification results were even worse than the detection results. Without a threshold to judge the "brightness" of an oil droplet in epi images, the epi was essentially rendered useless. Misclassification between the principle, colorless and transparent types was more common than not. In many cases the all the transparent oil droplets would be completely misclassified, or not even detected at

all. Figure 3.2 provides a look at the classification short comings of the old method in addition to some detection errors.

The failures of the initial approach led to the use of deep learning and CNNs. CNNs offered the best possible option to classify the oil droplets in the retina images. CNNs have also found footing in the fields of detection. With this in mind it was determined that CNNs could provide the best classification and detection if trained properly with enough data.

3.2 CNN

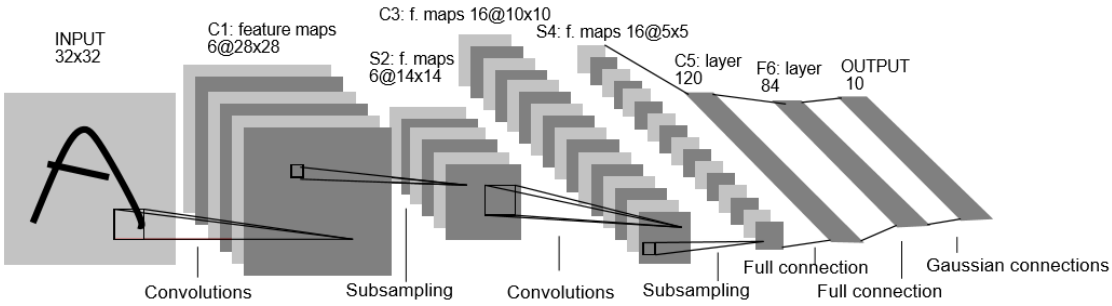


Figure 3.3. Visualization of the LeNet architecture from (LeCun et al., 1998) that is used in this thesis for classification

In order to produce the best possible classification results the pre-classified data of the avian oil droplets was used to train a CNN. Having large quantities of pre-classified data made moving to CNNs a logical next step for classification. However, even though the data was pre-classified it needed to be extensively formatted to be used for CNN training.

Most CNNs follow the same basic structure, multiple convolutional layers followed by fully connected layers (LeCun et al., 1989; Szegedy et al., 2014). A convolutional layer is a feature extractor that contains multiple feature maps (LeCun & Bengio, 1995). For larger data-sets, larger layers or more layers will

usually be added to the CNN (Lin, Chen, & Yan, 2013). Figure 3.3 gives the standard architecture for a CNN from (LeCun et al., 1998). The modern day CNN format described by (Krizhevsky et al., 2012) consists of several convolutional layers, max pooling layers, three fully connected layers and lastly a loss layer. A CNN can use one of many loss functions, but standard CNNs tend towards using the softmax loss.

Training a neural network is in essence trying to minimize the loss for the entire data set using back propagation. With back propagation the weights at each layer are adjusted and the gradient for the loss is then computed (LeCun, Kavukcuoglu, Farabet, et al., 2010). Training adjusts the weights of each layer in the network to minimize the loss. The following sections describe the process of training a CNN and the multiple ways it was used to classify photoreceptors, and assist in detection.

3.2.1 Caffe

The Caffe framework by (Jia et al., 2014) is used to create and train all the CNNs in this paper. Caffe is an open source deep learning framework created by the Berkley Vision and Learning Center. Caffe is considered to be the lead framework for deep learning, providing an open and flexible framework that is supported by a dedicated team at BVLC and a large open source community of developers. Caffe is functional on both the CPU and GPU, and is also integrated with Nvidia's CUDA Deep Neural Network Library (cuDNN). Caffe is implemented in C++ and is functional in Python and Matlab.

Caffe builds networks layer by layer, where each layer of a network is described in Google Protocol Buffer format. Caffe defines networks in bottom up format where the data layer is considered the bottom and the top most layer is generally the loss layer. Data that is passed through a Caffe network is stored in blobs (Jia et al., 2014). A blob is a representation of an N-dimensional array, where

N is the batch size, that not only hold images, but other data like network parameters.

As previously stated networks in Caffe are built layer by layer. The most important tasks of each layer are the forward pass and the backwards pass. The forward pass takes a blob as input, computes the output and passes it to the top. A backwards pass computes parameter gradients from output gradient that gets passed to it. A blob that is passed through the entire network simply goes through a series of forward passes and then backwards passes.

3.3 Classification

Using LeNet CNN architecture a network was trained to classify images of single oil droplets. This network serves two purposes to this thesis. First, it acts as a proof of concept that CNNs can individually classify one oil droplet type from another. Second, the CNN created in this section is used for cell classification in the image adjusted detection described later on. With the exception of the R-CNN network created later, this CNN is the primary method used for retinal oil droplet classification.

3.3.1 Training Data Augmentation

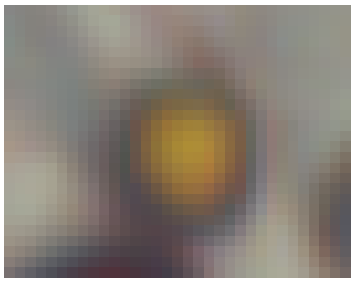
To format the data to create a CNN for single cell classification a large amount of images containing a individual cells would need to be cropped from whole images. To reduce the time it would take to manually cut and classify thousands of individual images for each type of oil droplet all cropped images were augmented. For each image that was cropped eight images would be labeled. These images were produced by rotating the image by 90 degrees three times, and then color adjusting the image and rotating it again by 90 degrees three times. It was found that using a low end vector of [0.1, 0.1, 0.0] and a high end vector of [0.6, 0.7, 1] adjusted the

image color enough to be variable from the original while not affecting the training results.

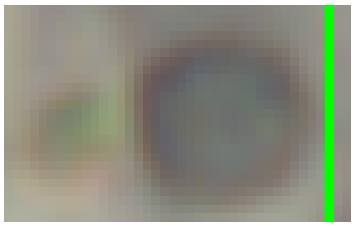
Each augmented image is labeled with a number 1 through 5. The number corresponds to the classification of the image. The label for each oil droplet is shown in table 3.1. The same process of cropping, augmenting and labeling is repeated to obtain data for testing, but significantly less data is used. Table 3.1 presents the quantity of data used for training the CNN. Figure 3.4 shows a yellow oil droplet and principle oil droplet image that were both used to train the single cell network.

Table 3.1
Single cell image counts

Type	**Label**	**Train**	**Test**
Red	1	1632	210
Yellow	2	1517	165
Transparent	3	420	65
Colorless	4	790	100
Principle	5	1663	155



(a) Yellow oil droplet image used for CNN training.



(b) Principle oil droplet image used for CNN training.

Figure 3.4. Images like these are labeled with the corresponding class number for training the single cell network.

3.3.2 CNN Training

Our CNN is created using the standard LeNet architecture (LeCun et al., 1998). To standardize the training and testing data sets the network crops the images to 24x24. After all the images are re-sized to 24x24 the training and testing image sets are compiled into two separate lmdb files (Chu, 2011), one for the training data set and one for the testing.

For the training lmdb the image mean is computed. An image mean is a 24x24 image that is created by computing the mean value at every pixel location between all the photos in the lmdb. Before an image is passed through the network it has the mean image subtracted from it. Subtracting the image mean helps with feature extraction during the images forward pass through the network. Subtracting the image mean eliminates commonalities that exist across some or all classes, leaving only the distinct features for each image for training or classification. Figure 3.5 shows the image mean computed for the single cell network described in this thesis

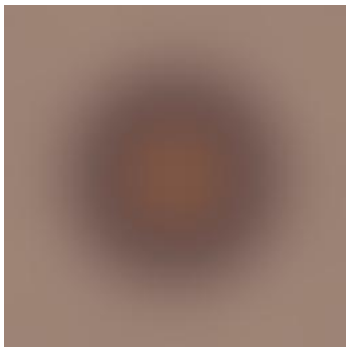


Figure 3.5. Image mean calculated for single cell network

With the images cropped, lmdb's created and image mean computed the network can begin training. The single cell network trained for 7,530 iterations on an Nvidia Tesla k40 GPU. Training took 40 minutes and 21 seconds to complete with a reported accuracy of 0.8757 on the test set and a loss of 0.0559.

3.3.3 Oil Droplet Classification

Before images can be passed through the network and classified they must be modified to meet network and Caffe specifications. The image must have its color channel switched from RGB to BGR. The resulting image is in W x H x 3 format where 3 represents the now BGR color channels. The use of BGR instead of RGB is part of the Caffe frameworks. After the channels for the image have been permuted the mean is subtracted for the the image and the results is passed through the network.

For each image that makes a forward pass the network will output the scores for each class in the network. The scores represent the probability that the object in the image is of any one type. The possible values for a score are between 0 and 1. The scores are not independent of each other meaning that the sum of all the scores must equal 1.000.

$$\sum_{i=1}^{n} score_i = 1.000 \tag{3.1}$$

Generally the image is classified based off the label with the highest score, known as the max label. Since the its possible for the max score to be less than 50% its not always worth while to just output the max label. In our application the score for all five classes are output sorted from greatest to smallest.

3.4 R-CNN

R-CNN is a special type of CNN that is specially designed for detection. Originally created by (R. Girshick et al., 2014) it has since been upgraded to fast R-CNN (R. Girshick, 2015) and more recently faster R-CNN (Shaoqing Ren, 2015). R-CNN uses the ImageNet CNN implemented in Caffe and extracts a feature vector for each region of interesting in an image. Training an R-CNN is done through network fine tuning. This involves taking a network that was previously trained for classification, and then training it for detection. Fast R-CNN and faster

R-CNN can also fine tune deeper networks like VGG16 (R. Girshick, 2015; Shaoqing Ren, 2015).

The network architecture for the original R-CNN is unchanged from the ImageNet CNN, but implementations of fast R-CNN and faster R-CNN have made changes. Fast R-CNN, which this thesis uses, defines two separate loss layers, a softmax loss and a smooth loss. Both loss layers branch from a fully connected layer and are separate from each other. The softmax loss computes the loss associate with the classification score for a proposal. The smooth loss computes the loss for a bounding box prediction. Loss is a term used to define the error in classification.

All versions of R-CNN are implemented in C++ as a branch of the Caffe repository. The original R-CNN and fast R-CNN are implemented in Python as well, while faster R-CNN was originally implemented in Matlab. Fast R-CNN also has basic Matlab function-ability, but this thesis uses the Python implementation. Training the fast R-CNN network for oil droplet detection follows similar steps as with the standard CNN created previously, but with some minor additions. These additions are creating bounding box annotations for each image, and computing regional proposals.

3.4.1 Training Data Augmentation

To label data to train an R-CNN network each whole image needs to have an annotation file associated with it. The annotation file contains bounding box coordinates and labels for each of those bounding boxes. Using previously classified images for reference, bounding boxes were places around all the classified cells. Each bounding box is labeled as either red, yellow, transparent, colorless or principle. The bounding box and the label are saved to a .xml file that can be read by Caffe during training. Internally bounding boxes are represented as an (x, y) coordinate with a height and width. The (x, y) coordinate is the top left corner of the box on the image. Annotations for each image were obtained using the

trainingImageLabeler tool provided in the Matlab Computer Vision Toolkit. Figure 3.6 provides a look at image labeling using the trainingImageLabeler. Even with previously classified data the process had to be done by hand. This effected the over all amount of training data that was processed.

To train the R-CNN network bounding box proposals must also be sent in as input (R. Girshick, 2015). Bounding box proposals are a set or regions in the image where an object might be. The network makes final decisions as to whether a proposal is the background or an object. In this thesis object proposals for each image are generated by running a selective search (Uijlings et al., 2013), but we recognize the existence of other algorithms for generating object proposals. Selective search uses a combination of image segmentation and exhaustive search. This produces a set of all possible object locations in the image. These possible locations are the object proposals. Selective search can produce thousands of proposals for a single image. The majority of these are false positive detections. Using classification these detections can be identified as the background or a possible oil droplet. The selective search saves all the object proposal bounding boxes to a single .mat file associated with the training data set.

3.4.2 R-CNN Training

The fast R-CNN framework created by (R. Girshick, 2015) was used to train and test our R-CNN network. To train a fast R-CNN network for cell classification and detection the Caffe ImageNet 2014 model was fine tuned. The fine tuning added the cell classes to the existing model over 40,000 training iterations. Originally the network was trained using the same images as the single cell classification CNN. The resulting network posted poor results for whole cell regions, never being able to classify an ROI with a score above 0.25. The underlying issue was that features of one ROI relevant to any others in the image were nonexistent since only one ROI existed per image. To combat this issue the network was

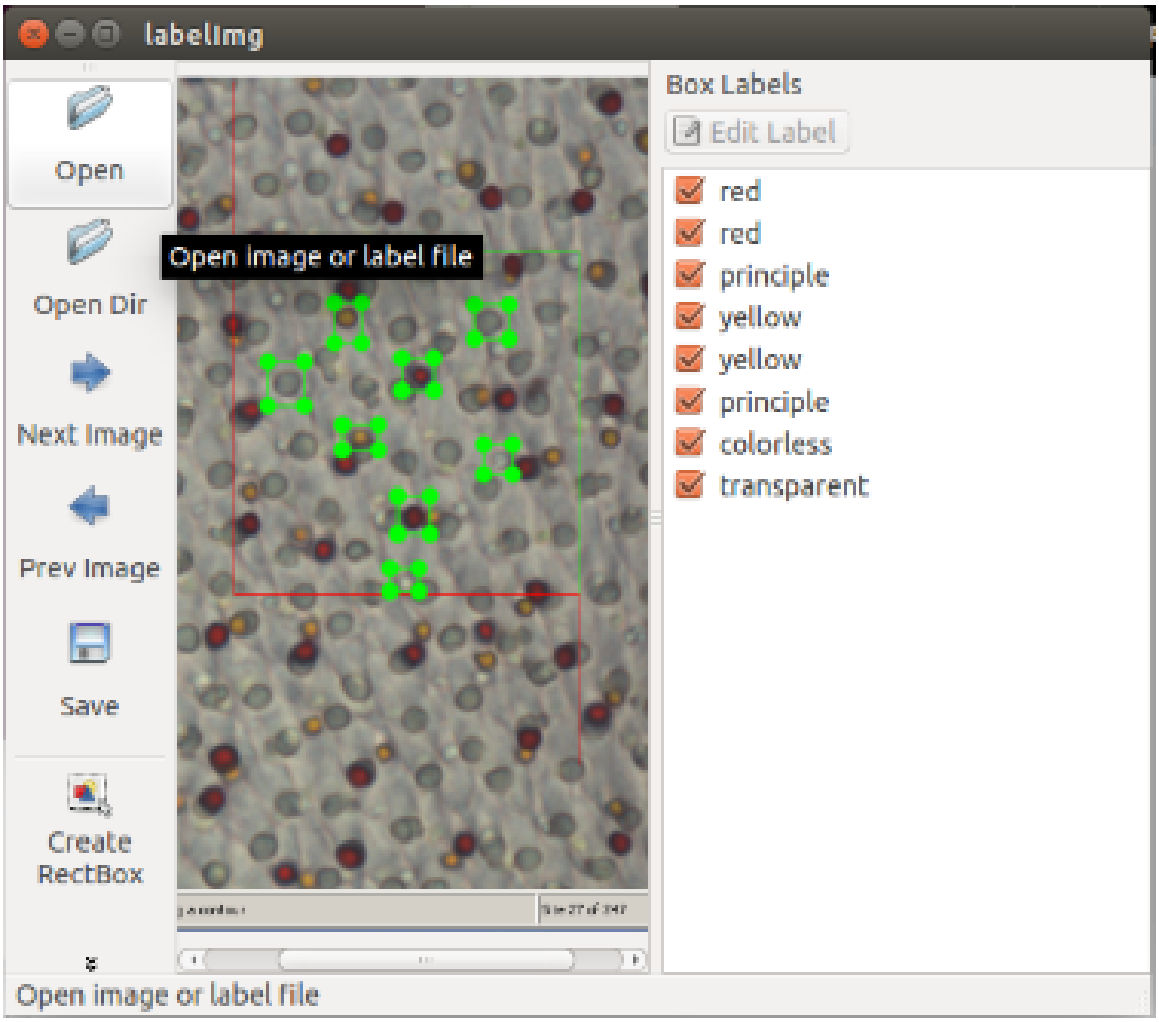


Figure 3.6. ROI annotations for a retina cell region.

re-trained using images of entire retina regions. This allowed the network to better recognize the placement of features relative to one another. For the training data set the object proposals are stored in a single .mat file produced by Matlab.

Similar to training for the single cell network the images need to be compiled into an lmdb before training the R-CNN network. Both a training and annotation lmdb are be created. The annotation lmdb is similar to the training lmdb except it encapsulates all the annotation files. Since the ImageNet 2014 model was fine tuned

a test lmdb was not needed. Similarly, because fine tuning was performed and image mean was not calculated for the network either. To fine tune the network Caffe is given the ImageNet 2014 model, its solver state and the lmdb for the training image as input.

Training the R-CNN over 40,000 iterations took significantly longer than the single cell CNN. Total training time took close to five hours. The final reported loss was 0.1459. Accuracy results for R-CNN were not output during training.

3.4.3 R-CNN Detection

The fast R-CNN network runs images in batches. The total size of the batch is describe by a text document that lists the absolute paths to all the images in the set. For each image that is going to be processed object proposals have to be found. The object proposals are found identically to the training set by using the selective search. Using selective search approximately 2K object proposals are created for each image. Unlike the training data set, however, the object proposals are not stored in a single .mat file. This is because images with in the batch are passed through the network individually. Figure 3.7 shows the results of a selective search. All the regions surrounded by a red bounding are object proposals generated by the selective search. These are the regions that will be classified as one of the oil droplet types or the background.

After the image does a forward pass through the network the list of object proposals and scores for each proposal are returned. For each bounding box the array of scores associated with it is sorted from maximum to minimum. Now that every box has a max score associated with it ROIs that are considered misses are filtered out.

ROI filtering consists of multiple steps. The first step is to discard all the bounding boxes whose max score is not for one of the five oil droplet types. Theses ROIs are considered to be background regions. The next step in filtering out the

background regions is determining a max score threshold. This defines a max score a non-background classified ROI must have to be considered a positive detection. In the classification CNN the max score threshold was not used because all images contained an oil droplet. For R-CNN the threshold must be defined to further eliminate misclassified background regions. After testing it was noted that a threshold of 0.85 preformed well for the eliminating miss classified background ROIs.

After the background regions are filtered out overlapping detections need to be eliminated. Bounding boxes that are bigger the 40x40 are eliminated first. These bounding boxes are too big and likely will contain multiple cells. Next all the remaining overlapping bounding boxes of the same classification must be removed. In the review of the Poisson distribution is was stated that oil droplets of the same class can only occur so close to each other. To remove overlapping bounding boxes Non-Maxima Suppression (NMS) is used (Devernay, 1995). Given a threshold defined as the maximum amount of overlap two bounding boxes may have, NMS finds all the groups of overlapping bounding boxes and removes the boxes with the lowest score. This process assumes that the bounding box with the highest score is the most centered around the object.

After all the ROIs for each oil droplet type are filtered and removed the remaining ROIs are assumed to be positive detections. To improve the readability of the data the results are printed to five files. Each file contains the input image and all the labeled bounding boxes for one type of oil droplet. Each bounding box in the output image is labeled with the score for the oil droplet.

3.5 Adjusted Image Detection

During the process of augmenting training and testing images for the classification CNN it was observed that the [0.1 0.1 0.0, 0.6 0.7 1] block vector used to adjust images actually made the cells more visible from the background, see Figure 3.8. Previous non-machine learning methods for cell detection performed

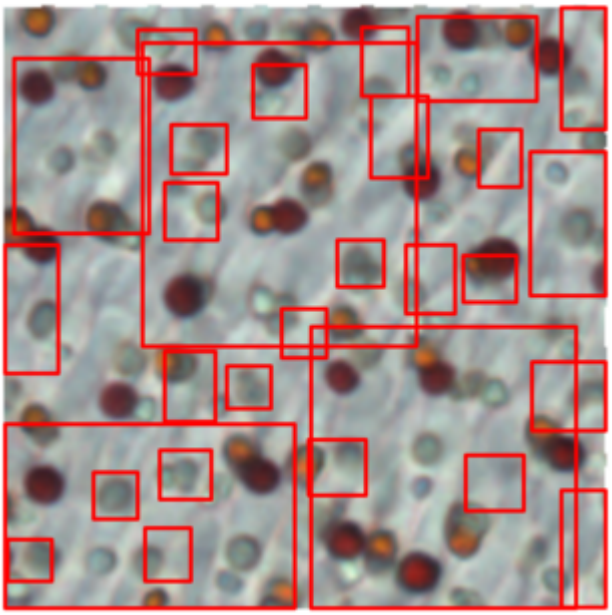


Figure 3.7. Selective search ROIs for a retina cell region

poorly because in most cases the cell could not be segmented from the background. With greater contrast between the oil droplets and the background coupled with a CNN for classification cells could now be detected with greater accuracy while also providing CNN classification. Using the Computer Vision Toolkit provided by Matlab first a detection method that coupled color-based image segmentation with k-means and boundary tracing was implemented. Then a method using the two stage Hough transformation for circle detection was implemented. The adjusted image is strictly used for detection. CNN classification was performed using the unaltered image using each of the regions detected by the algorithm.

3.5.1 K-Means Segmentation with Boundary Tracing

The first step to segmenting the retina cell regions with k-means is converting the image from RGB to Lab. Lab is a color space where L is the luminosity, a is represents a colors location on a red-green scale and b is a colors location on a blue-yellow axis. Color information is now stored in a and b. The benefit of being in

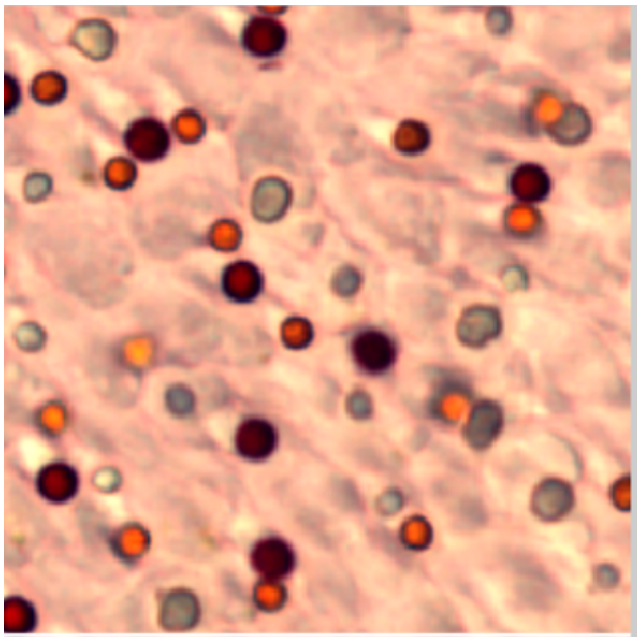


Figure 3.8. Block vector adjusted image. The cells tend to be more distinctive after adjustment.

Lab space is that Euclidean distance can be used as the distance metric for k-means. In Lab space the difference between two colors is their Euclidean distance. Euclidean distance is the straight line distance between two points.

Our k-means algorithm splits the input image into three clusters, background, red and yellow, and principle, colorless and transparent. We utilize the fact that the distance between red and yellow is relatively small compared to their difference from the rest of the image. The same notion also holds true for the principle, colorless and transparent cells who all appear as a shade of gray. In Matlab a random seed is used during the k-means computation. This seed represents the initial points chosen as the center of each cluster. With the seed, or center, being random the clustering results might always be different on a single image. This can be a problem for replicating results. To solve this the seed is manually set in our algorithm so that the results for one image will always remain the same.

After clustering, a matrix containing the cluster ID for each pixel in the image is created. While retina regions can contain a large number of pixels clustered into one of the clusters that contain the oil droplets, they should never outnumber the amount of pixels clustered into the background. After obtaining the background cluster ID by taking the mode of the ID matrix, all the pixel labeled in the background cluster have their RGB value set to zero as shown in Figure 3.9.

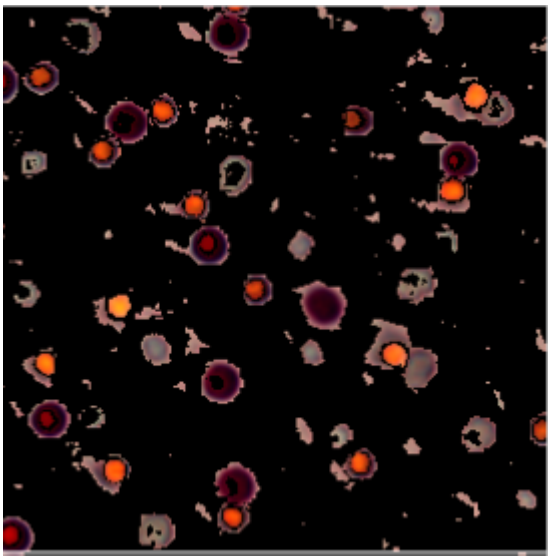


Figure 3.9. Pixels clustered into the background have their RGB values set to 0 in the original image.

After clustering, boundary tracing is run to segment the oil droplets in the image. The first step is converting image shown in Figure 3.9 to a binary image. For each non-black region in the image Moore neighborhood contour tracing is used to trace the boundaries(Gonzales, Woods, & Eddins, 2004). Starting at a black pixel Moore neighborhood contour tracing traverses all the neighboring pixels until a black pixel is found (Pradhan, Kumar, Agarwal, Pradhan, & Ghose, 2010). This process is continually repeated until the starting pixel is visited a second time. The set of visited pixel is then assumed to be the boundary of the oil droplet, Figure 3.10.

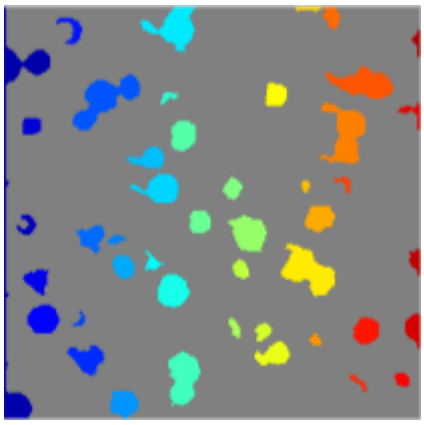


Figure 3.10. Boundary tracing using Moore neighborhood on segmented regions.

To classify each of the segmented regions they must be cropped out of the original image and passed through the network. To ensure the whole oil droplet is contained in the crop the center of the boundary is computed by the centroid. A 24x24 mask of the image is then made around this center point on the unaltered image and passed through the CNN. The center point is then marked with the resulting max label as shown in Figure 3.11

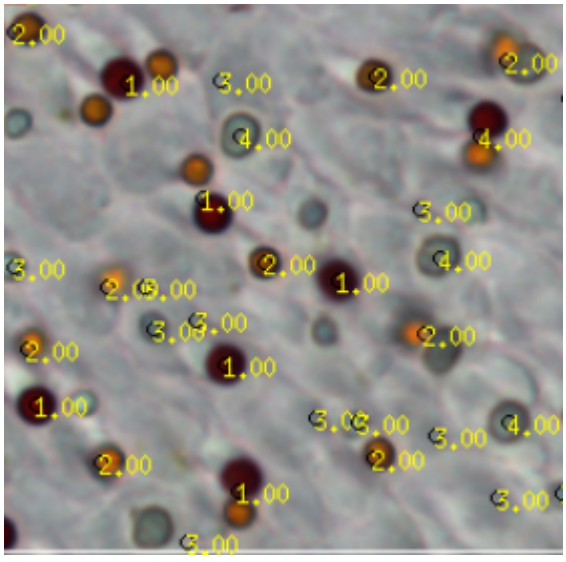


Figure 3.11. Classified photoreceptors detected using k-means clustering and boundary tracing.

3.5.2 Two Stage Hough Transformation

The third detection method implemented in our system is a two stage Hough transformation on an adjusted image. Hough transformation is a well know method for circle detection (Atherton & Kerbyson, 1999; Yuen et al., 1990). After image adjustment there is greater contrast between the oil droplets and the background of the retina region which makes edge pixels more identifiable. Ex

When using a Hough transformation for circle detection the polarity of the objects being detected must be defined. This helps with identifying potential circles and vote tallying. Looking at figure 3.8 we can see that, after adjustment, the oil droplets are generally darker than the background.

Computation of the two stage Hough transformation is performed in Matlab by the built in *imfindcircles* function. The output from the function gives a list of circle centers and their radii. For a visualization of the results see Figure 3.12. The phase-coding method of Hough transformation was also tested but produced worse results when compared to the two-stage Hough transformation.

After detection oil droplets are cropped from the original image, but in this case a *r+4 x r+4*, where r is the circle radius, crop is made. Four is added to the radius to pad the crop in case the center is not the true center of the oil droplet. The cropped images are passed through the CNN. The detected circle associated with the cropped image passed through the next work is labeled with the max score, Figure 3.13.

3.6 Summary

This chapter outlines the steps taken to train both the single cell and R-CNN networks, classify images and detect cells. The initial approach to the problem was first presented and its failures outlined. Data formatting and labeling for both the single cell network and R-CNN network is covered, including the image annotations required for R-CNN. Training methods and the differences between the

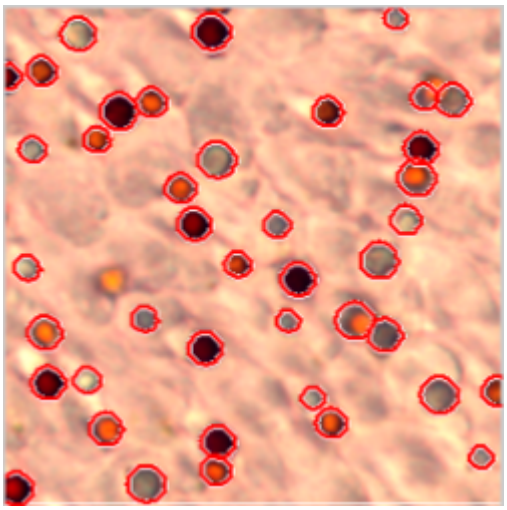


Figure 3.12. Oil droplet detection on the adjusted image using a two stage Hough transformation.

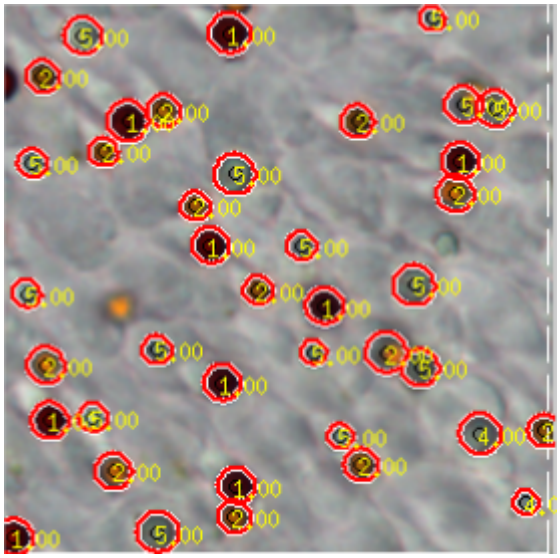


Figure 3.13. Classified oil droplets detected by two way Hough transformation.

two networks are noted. For R-CNN a brief overview of the selective search used to generate ROIs is covered. The methodology for processing images in both networks is describes. Steps to preparing the single cell data in Matlab is given, and filtering techniques to achieve accurate results for the R-CNN network are presented. Lastly, two non-machine learning detection algorithms were implemented on an adjusted

image. The detections from both the k-means segmentation and two-stage Hough transformation are passed to the CNN for classification. In the next section the results will be discussed in detail.

CHAPTER 4. RESULTS

All the results produced by the algorithms used are compared to ground truth images that have been pre-classified, and verified for their correctness by biologist from the Purdue University Biological Sciences department. Each CNN was tested using data with an identical format to the data that the network was trained and tested with. The ground truth images that the test results are compared with also have the same format as the images used to train and test each network. Any pre-processing that had to be done to format the images before testing is not considered as part of the tests or study. The results for the tests were viewed on a per cell type basis and then brought together to evaluate the entire network.

All tests are performed on a computer running Ubuntu 14.04 using and Intel i7 processor, 8 GB of RAM and an Nvidia K40 graphics card.

4.1 Classification

New single cell images were cropped out of ground truth pre-classified images that were not used for training. The images were re-sized to meet the CNNs requirements. Table 4.1 shows the accuracy for the classification of each cell type. For testing a total of 197 red, 166 yellow, 112 transparent, 143 colorless and 156 principle images were used. The resulting classification for each input image is the cell type with the highest score for that image. Even if the highest score was less than 50% that image would still get classified into the highest score.

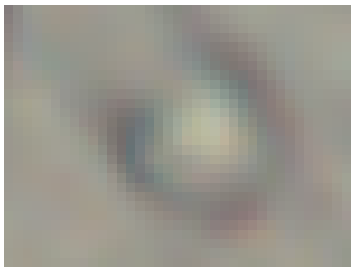
Looking at the results from Table 4.1 we can see that all the cell types perform well for the single cell classification with transparent being the only type to not obtain 90% accuracy. As Table 4.2 shows the incorrect classification with the highest score for transparent was colorless. All 18 incorrect classifications for the

transparent tests were classified as colorless, and indeed the opposite holds true for colorless. All together we can see that across all the test the CNN obtains 95.09% accuracy, which is well within acceptable human range for cell classification.

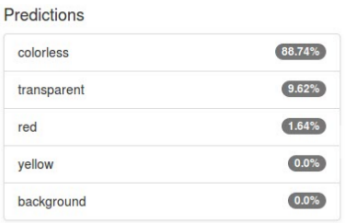
Our system does not incorporate the epi image into classification. The epi image is used to help classify between the transparent, colorless and principle types. Without any need for the epi image, however, we can see the a CNN can classify at a level acceptable to human classification. The lack of an epi image is the most likely result for the misclassifications between the colorless and transparent types. Figure 4.1 gives a sample of the output for the single cell network.

Table 4.1
Single Cell Network Classification Results

Type	**Images Used**	**Correct**	**Incorrect**	**% Correct**
Red	197	196	1	99.49%
Yellow	166	163	3	98.19%
Transparent	112	94	18	83.92%
Colorless	143	130	13	90.90%
Principle	156	152	3	98.07%



(a) Colorless oil droplet passed through the CNN for classification.



(b) Score outputs for colorless cells in 4.1(a).

Figure 4.1. Sample output for the results of a red oil droplet classification.

Table 4.2
Single Cell Network Score Extremes

Type	**Max Correct**	**Min Correct**	**Max Incorrect**	**Max Incorrect Type**
Red	100%	90.09%	69.99%	Yellow
Yellow	100%	64.24%	95.28%	Principle
Transparent	96.90%	49.94%	66.99%	Colorless
Colorless	99.91%	50.91%	73.26%	Transparent
Principle	100%	57.61%	94.49%	Yellow

4.2 R-CNN

The R-CNN network is a culmination of the classification and detection problems. Using fast R-CNN (Shaoqing Ren, 2015) selective search finds regions of interest to be classified. Because there is no guarantee that any of the ROIs will have a cell in it fast R-CNN uses classification for detection. Detection is evaluated by the number of cells that are left undetected. Due to the nature of R-CNN false positive regions actually correspond to miss classified background regions. Classification is measured by the amount of misclassifications, similar to the single cell CNN.

4.2.1 Timing

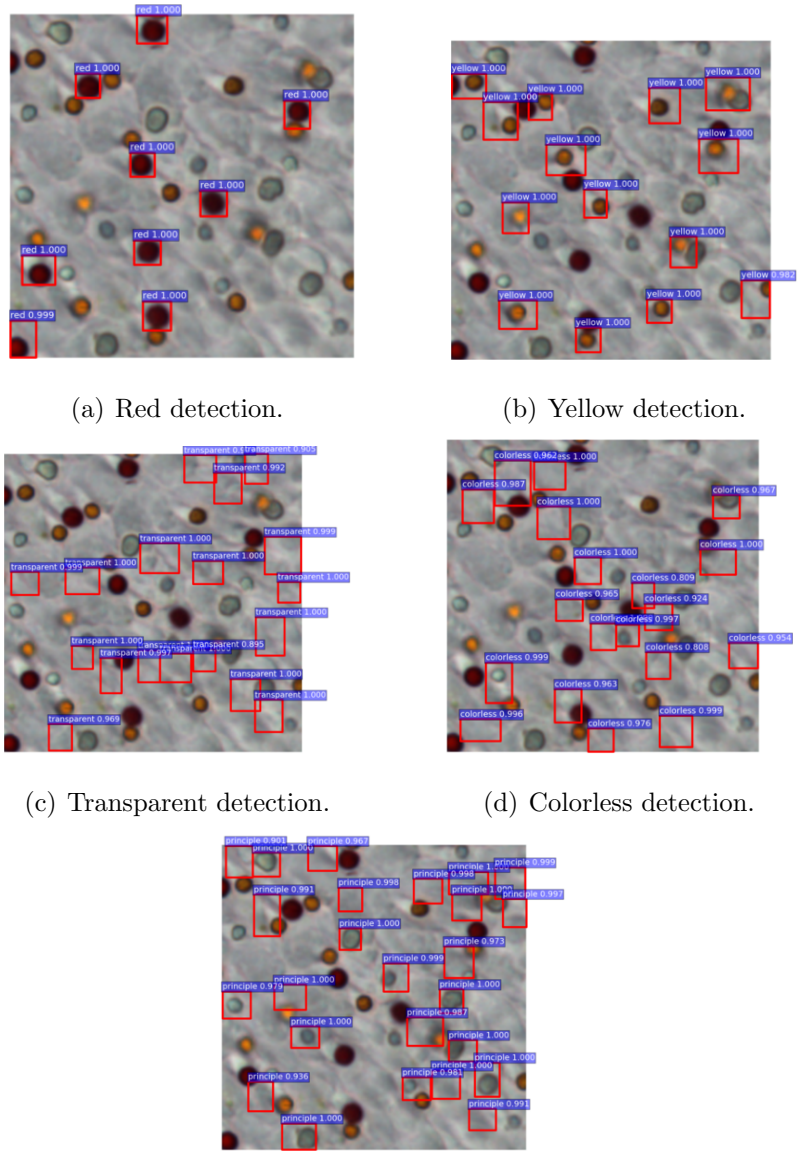
The fast R-CNN network is fast at classification and detection. On average it can process a full size retina image in 0.267 seconds. Processing an entire retina of 211 images takes 1.13 minutes. The results for each oil droplet type are saved into separate files as shown in figure 4.2 to make the results for each type more view-able.

4.2.2 Detection

Running the selective search on an images produces thousands of ROIs thanks in part to the noise many of the retina cell photos contain in addition to the large number of cells. When feeding the ROIs into the network regions that get classified as background are discarded. In addition only regions whose score exceeds 0.85 are considered to be classified. This threshold was chosen to account for large ROIs that might contain many cell but produce a high score for the most prominent cell in that region.

Detection is still an open problem, (R. Girshick et al., 2014), and the results from the fast R-CNN network show just that. The coverage for the red and yellow oil droplet combine across all tests is above 95%. These are the most distinguishable cells and can easily be found with the selective search algorithm. Principle also tends to perform well. As Figure 4.2(e) shows all the principle cells in the image are covered. Issues do arise in images where the principle cells are more faded and less defined.

Transparent and colorless cells are where the network does not perform as well. Figures 4.3(a) and 4.2(d) show that in many cases the background is classified as a transparent or colorless type while failing to detect the actual cells in many cases. In many cases the cause of these missed detections was because the bounding boxes that covered those cells were to large and removed. ROIs that are larger than 40x40 are also removed and not considered. This is done to avoid repeated detections and reduce the number of false positives. It was observed that allowing large ROI would cause more false positives to appear than the number of misses that smaller ROIs would cause. This was the main factor in deciding whether to include large or small ROI.



(a) Red detection.

(b) Yellow detection.

(c) Transparent detection.

(d) Colorless detection.

(e) Principle detection.

Figure 4.2. Fast R-CNN detection results.

4.2.3 R-CNN Classification

Because the fast R-CNN network is different in many ways from the single cell network the classification results do differ. As figures 4.2(a) and 4.2(b) show the classification for all the detected yellow and red oil droplets is around 100%. Across all test 97% of red and yellow cells were properly classified and detected. Errors

would only occur in the event a yellow cell appeared more orange than yellow. These cells could end up being classified as red instead of yellow. The classification scores for the red and yellow oil droplets tend to remain around 1.000 meaning the network is very confident of the classification. Principle oil droplet detections also tend to perform well, reach mid 80%'s to low 90%'s for detection on most images.

As previously stated in R-CNN classification leads into detection which can exacerbate the results of a false classification. Any classification that the network believes to be relatively good, has a high score, the network will keep. This is how false classification arise in R-CNN. The primary victims of this phenomenon are the transparent and colorless cells.

Looking at Figure 4.3 we can see that both the transparent oil droplet and the colorless oil droplet in the photos can easily blend in to the background. This can lead to one of two possible errors. The first as discussed is false positive, where a background image is actually positively classified as a transparent or colorless cell as shown in Figures 4.2(d) and 4.2(c). The other situation is a transparent cell or colorless cell gets classified as background. This scenario would cause that specific detection to be filtered out with all the other background images, leading to a missed detection.

(a) Transparent oil droplet. (b) Colorless oil droplet.

Figure 4.3. Transparent and colorless oil droplets tend to blend with the background.

4.3 Adjusted Image Detection

After image adjustment the oil droplets became more distinguishable from the background. Using classical segmentation methods from computer vision oil droplets were detected with a higher accuracy than even the R-CNN network produced. Not only was the detection better but it eliminated the need to train hundreds to thousands of images with bounding box proposals, and the previously created single cell CNN could be reused to classify the detections. The first implementation is a Lab color k-means segmentation with boundary tracing. The second is circle detection using two stage Hough transformation.

The same CNN was used for classification for both detection algorithms which lead to consistent classification results between the two. The classifications prove to be very effective for the detected oil droplets, posting results similar to the classification results shown in the CNN black box testing. Their is still confusion between the colorless and transparent cells, but the principle, red and yellow classifications are consistent with those posted by the CNN.

4.3.1 K-Means Detection

K-means clustering with boundary tracing produces good detection results for retina region image with quite backgrounds. For images that could be segmented with k-means on average 8.5 oil droplets were missed among an average 38.25 oil droplets per image. False positive detections occurred occasionally when small segmented regions would be classified at a transparent or colorless type. Adjusting the image vastly improved the quality of k-means segmentation, where before it was generally unusable for most images. Even with the improvement, however, k-means was still unstable for many images.

Similar to issues with R-CNN the downfall of k-means segmentation is the varying image quality for the input images. For many test images k-means works well segmenting the cells, shown in Figure 4.4, but for images with more noisy

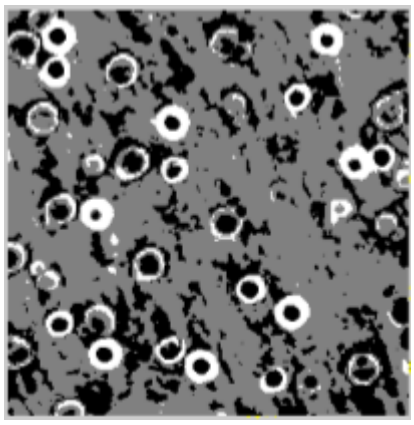
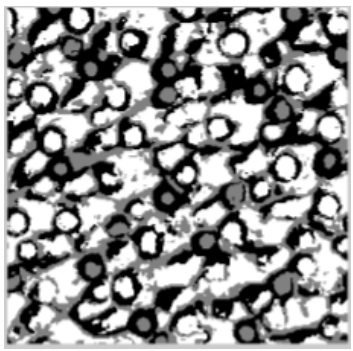
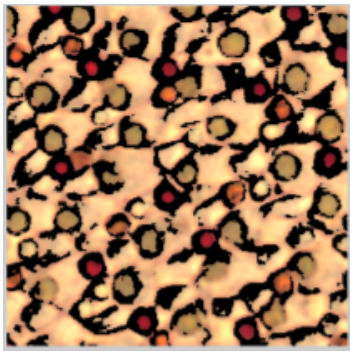


Figure 4.4. A retina region that is well segmented by k-means. Background regions clustered in gray where cell regions comprise of the black and white clusters.

backgrounds the results are not as good. There are many cases where oil droplets get clustered with the background like what is shown in Figure 4.5(a), and can produce a segmented image like the one shown in Figure 4.5(b). Because the background has not been fully eliminated from the image boundary tracing does not segment out oil droplets for classification.



(a) The white regions cover both the background and oil droplets.



(b) The segmented image still contains a majority of the background.

Figure 4.5. With a noisy background k-means will segment the image poorly.

4.3.2 Hough Transformation Detection

Many of the inconsistencies that arise during k-means clustering have little effect on the two stage Hough Transformation. Across the majority of testing images circular oil droplets were detected at a high rate. On average only 2.0 oil droplets were missed in a cell region. False positives were also more uncommon with Hough transformation compared to k-means with approximately of 1.4 false positives per-image.

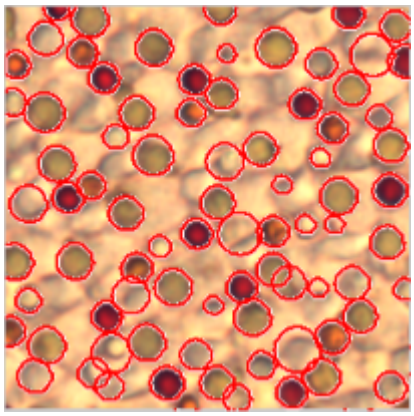


Figure 4.6. Using two way Hough transformation on an image k-means could not segment.

Across all the images tested the Hough transformation was consistently able to detect the oil droplets. In images that failed for k-means like the one showed in Figure 5.1 Hough transformation detected all or the majority of the circle, shown in Figure 4.6, but occasionally with the inclusion of more false positives. Instances of missed detection occurred in more cases for yellow or transparent oil droplets, when the polarity of the cell was lighter than the background, Figure 4.7 The two stage Hough transformation used in this thesis searches for circles darker than the background. This can be solved by reducing the image detection threshold defined by the algorithm, but will also result in more false positives.

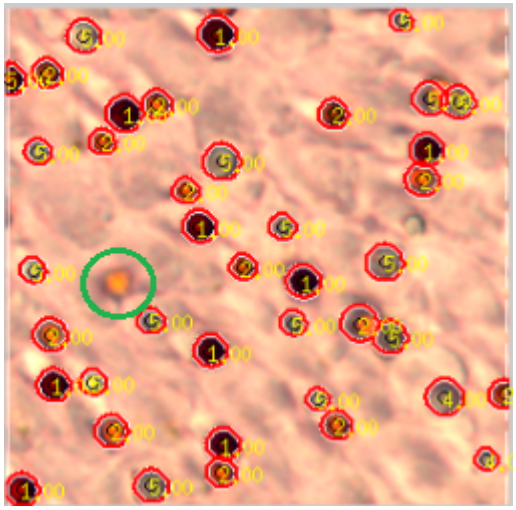


Figure 4.7. A yellow type oil droplet is left undetected due to its lack of contrast with the background.

4.4 Summary

In this chapter the results for the classification and detection algorithms are presented. First we show that a CNN can classify oil droplets at or above 95%, which represents human effectiveness. The CNN trained on the five oil droplets classified the red, yellow and principle types test data in the upper 90%. Classifying the transparent and colorless type was not as accurate but still showed effective classification results by classifying test data at 84% and 91% respectively. The results show that even without the use of an epi image, which before was thought to be critical to classification, a CNNs trained on single cell images could classify at a human level. The effectiveness at classification is also compounded by the speed at which large amounts of data can be classified at once.

Second, the results of the fast R-CNN network were reviewed. While detection is still known to be a very open problem in computer vision the fast R-CNN produced promising results. Red, yellow and principle type cell all posted promising detection results. The results for transparent and colorless detection were not as good, shown in Figures 4.2(c) and 4.2(d). Much of the inaccuracy can be attributed to a lack of training data, specifically background training data.

Lastly, the results of non-machine learning detection and segmentation algorithms on adjusted images are reviewed. The results for k-means segmentation with boundary tracing provide good results for some images, but not all. While it posts low rates of false positives, a lack of consistent seeding and cells still blending with noisy backgrounds reduce the effectiveness. There is no way to know the most optimal seed for k-means for anyone image. Using a random seed would make results inconsistent for the same image, and no one seed will be optimal for every image. With no way to know the optimal seed for every image, the results are inconsistent from one image to the next.

Two stage Hough transformations results in less misses and false positives per image than k-means. Hough transformation also posts significantly more consistent results across all images. Regardless of the quality or contrast after image adjustment most oil droplets will be covered in the retina images. Misses from Hough transformation occur in instances where an oil droplet has a lighter polarity than the image background. Many misses can be found by increasing the detection threshold, but this would also increase the number of false positives. The classification results for both Hough transformation and k-means segmentation reflect the results posted by the CNN they use.

CHAPTER 5. CONCLUSIONS AND FUTURE WORK

In this thesis methods for automatically detecting and classify avian retina oil droplets, or oil droplets, are presented. Using labeled images of avian retina oil droplets we trained a CNN that can perform human level classification of avian oil droplets. Detection was first performed by training R-CNN for detection. After adjusting the retina region image using a [0.1 0.1 0.0, 0.6 0.7 1] block vector we showed that traditional detection and segmentation algorithms can be even more effective than R-CNN, while providing the classification benefits of the trained CNN. In this chapter we draw conclusions from the methods and results laid out in this thesis, and offer insights into areas for future work.

5.1 Classification

The results for the CNN trained for show that even without using the epi image oil droplets can be automatically classified at human level. While errors do occur classifying between the transparent and colorless types, overall accuracy for the network sits at around 90%. Even without being paired with automatic detection, the CNN trained in this thesis eliminates the need to train a human to tell the differences between cells. The possibility of speeding up the processing time for retinal cell regions would have a profound impact on avian research in biology.

5.2 Detection

Three detection methods were implemented in this thesis, fast R-CNN, k-means with boundary tracing and two stage Hough transformation. While from a computer vision perspective each of the methods worked well for detecting oil

droplets, only the two stage Hough transformation was effective from a biology perspective, producing on average 3.4 detection errors per 38.25 oil droplets. The two stage Hough transformation on an adjusted image was the only method that produced consistent and usable results across all images.

5.2.1 R-CNN Detection

Fast R-CNN was created such that a single CNN would be trained to combine both classification and detection. While the R-CNN detects and classifies the red, yellow and principle type very accurately, detection and classification for the transparent and colorless types was practically nonexistent. No longer are transparent and colorless oil droplets just being mixed up, but the introduction of a background class leads to many not even being classified. Even worse background regions of interest can be misclassified as either a transparent or colorless type.

Fast R-CNN has proven to be one of the most if not most effective detection method available on the ImageNet detection challenge. With substantially more training data its still possible that either fast R-CNN, or its newer iteration faster R-CNN, could provide best case detection. Selective search produces many ROIs, but more ROIs also increase the chance for errors to occur.

5.2.2 K-Means Segmentation

Using image adjustment increases the effectiveness of k-means segmentation, but only for some images. Images that don't have smooth background will inevitably have either cells clustered with the background or the opposite. There are two primary causes for the shortcomings k-means segmentation. The first is that the seed that is set is not universal. The native Matlab implementation of k-means optimizes itself by choosing a random seed for the image. This seed represents the initial center for the clusters. To avoid different results for the same image the seed in manually set before hand. The same seed is not consistent for all the images, and

there is no way to know what the best seed for an image is beforehand without testing multiple seeds on a single image. The second and most important is that the same block vector used to adjust the image with k-means is not as effective for all images. For the majority of images the [0.1 0.1 0.0, 0.6 0.7 1] block vector is effective a increasing the contrast between the background and oil droplets, but as shown in Figure 5.1 there are many cases where transparent and colorless cells still blend in with the background leading to poor segmentation.

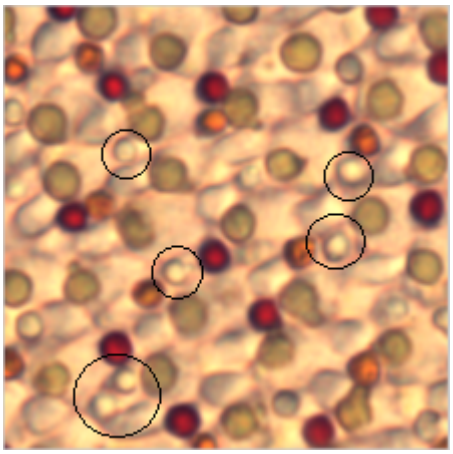


Figure 5.1. Notice how many of the lighter cells blend well with the background. k-means would have difficulty segmenting this retina region.

5.2.3 Two Stage Hough Transformation

The best detection method for oil droplet detection implemented in this thesis was using two stage Hough transformation with an adjusted image. Not only were missed detections rare with this algorithm, but so were false positives. Even for images that were to noisy for k-means segmentation, the Hough transformation could handle with ease.

With no human input the Hough transformation performs well. Tweaking the detection threshold for individual images based on their quality produces human

quality detection across all images. With this in mind processing an entire image would be reduced to thresholding. Instead of a researcher having to count cells and classify them, they can adjust the threshold based on image quality and let the two stage Hough transformation and CNN take care of the rest. This possibility would dramatically speed up avian retina research and be a major step for biology.

5.3 Neural Network Training

Many if not most neural networks use tens of thousands of images to train a reliable network. Both the CNN and fast R-CNN in this trained were trained over hundreds of images. All the images were formatted mostly by hand and then augmented. A lack in training data could explain many of the classification and detection errors in the single cell CNN and fast R-CNN networks. Due the the varying quality of the images its also possible that images whose quality drastically deviates from the images used for training could experience errors. Substantially more data could increase the effectiveness of the networks could increase dramatically.

5.4 Areas of Future Research and Study

While the networks do perform well with greater amounts of training data the new networks could be trained that perform substantially greater than the current ones. This would be especially important for detection. The primary issue in the missed classifications in the R-CNN network stem from mislabeled background ROIs. expanding the amount of training data used and including varying image qualities would reduce this error.

The single cell network created is identical to the LeNet (LeCun et al., 1998) architecture. Using a slightly deeper and more modern network like GoogLeNet (Szegedy et al., 2014) would help with both classification and detection. While the

problem does not call for a very deep network like VGG16 (Simonyan & Zisserman, 2014) it would benefit from a more modern architecture.

The algorithms presented are all automated and as a result produce error. While it is a natural goal of computer science to reduce the amount of error to near human levels, it is still not a case in all computer vision problems. Introducing minimal human assistance into the algorithms proposed would help speed up the time it takes to count cells while also reducing the error to a more acceptable level.

Reinserting the epi image into the detection and classification process would also help improve results. While the epi image wouldn't necessarily offer much for classification, except for colorless and transparent, it would greatly help with colorless detection which has shown to be a particular problem. Similar to the single cell network showing that classification can be performed with out the epi image, creating a network trained on just epi images may reveal that the principle and colorless cells do have distinct features in the epi the human eye cant catch.

The networks outlined in this thesis present a small sample of how CNNs can be used in microbiology. The ideas presented in this thesis can easily be expanded to other areas in biology. With adequate amounts of properly formatted data, even the networks in this thesis could be fine tuned to include other biological microbes. This would expand the breadth of cells the networks can classify, providing perhaps a more general purpose too to biologist.

Following the completion of this thesis a new R-CNN method was released called faster R-CNN (Shaoqing Ren, 2015). Faster R-CNN is an improvement on the Fast R-CNN architecture that is used in this thesis. Moving from the fast R-CNN network to the faster R-CNN network in addition to more training data could produce better results.

5.5 Summary

In this thesis we provided a look at the effectiveness of convoltional neural networks in classification and detection for avian retina oil droplets. The need in biology for methods that can automatically detect and classify oil droplets has been outlined. The previous work in the areas of biology, computer vision and neural networks was reviewed to show the tools and methods that are built upon. We trained and applied CNNs in new ways that have not been proposed to date. This thesis has shown CNNs and computer vision algorithms have the potential to aid biologist in oil droplet detection and classification, which is currently a time consuming process.

LIST OF REFERENCES

LIST OF REFERENCES

Abdel-Zaher, A. M., & Eldeib, A. M. (2016). Breast cancer classification using deep belief networks. *Expert Systems with Applications*, *46*, 139–144.

Abràmoff, M. D., Magalhães, P. J., & Ram, S. J. (2004). Image processing with imagej. *Biophotonics international*, *11*(7), 36–42.

Alon, U. (2003). Biological networks: the tinkerer as an engineer. *Science*, *301*(5641), 1866–1867.

Atherton, T. J., & Kerbyson, D. J. (1999). Size invariant circle detection. *Image and Vision computing*, *17*(11), 795–803.

Bakshi, S., Mehrotra, H., & Majhi, B. (2011). Real-time iris segmentation based on image morphology. In *Proceedings of the 2011 international conference on communication, computing & security* (pp. 335–338).

Bertozzi, M., & Broggi, A. (1997). Vision-based vehicle guidance. *Computer*, *30*(7), 49–55.

Bowmaker, J. (1980). Colour vision in birds and the role of oil droplets. *Trends in Neurosciences*, *3*(8), 196–199.

Chen, T.-W., Chen, Y.-L., & Chien, S.-Y. (2008). Fast image segmentation based on k-means clustering with histograms in hsv color space. In *Multimedia signal processing, 2008 ieee 10th workshop on* (pp. 322–325).

Chu, H. (2011). Mdb: A memory-mapped database and backend for openldap. *LDAP11*.

Cruz-Roa, A. A., Ovalle, J. E. A., Madabhushi, A., & Osorio, F. A. G. (2013). A deep learning architecture for image representation, visual interpretability and automated basal-cell carcinoma cancer detection. In *Medical image computing and computer-assisted intervention–miccai 2013* (pp. 403–410). Springer.

Davies, E. R. (2004). *Machine vision: theory, algorithms, practicalities*. Elsevier.

Devernay, F. (1995). A non-maxima suppression method for edge detection with sub-pixel accuracy.

Donahue, J., Hendricks, L. A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2014). Long-term recurrent convolutional networks for visual recognition and description. *CoRR*, *abs/1411.4389*. Retrieved from `http://arxiv.org/abs/1411.4389`

Duda, R. O., & Hart, P. E. (1972). Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, *15*(1), 11–15.

Duda, R. O., Hart, P. E., & Stork, D. G. (2012). *Pattern classification*. John Wiley & Sons.

Fernández-Juricic, E., Ojeda, A., Deisher, M., Burry, B., Baumhardt, P., Stark, A., ... Ensminger, A. L. (2013). Do male and female cowbirds see their world differently? implications for sex differences in the sensory system of an avian brood parasite. *PloS one*, *8*(3), e58985.

Fukushima, K., & Miyake, S. (1982). Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern recognition*, *15*(6), 455–469.

Girshick, R. (2015). Fast r-cnn. In *International conference on computer vision (ICCV)*.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 580–587).

Girshick, R. B., Donahue, J., Darrell, T., & Malik, J. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, *abs/1311.2524*. Retrieved from `http://arxiv.org/abs/1311.2524`

Gonzales, R. C., Woods, R. E., & Eddins, S. L. (2004). *Digital image processing using matlab*. Pearson Prentice Hall.

Guan, B. X., Bhanu, B., Talbot, P., & Lin, S. (2014). Bio-driven cell region detection in human embryonic stem cell assay. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, *11*(3), 604–611.

Gupta, S., Girshick, R., Arbeláez, P., & Malik, J. (2014). Learning rich features from rgb-d images for object detection and segmentation. In *Computer vision–eccv 2014* (pp. 345–360). Springer.

Hariharan, B., Arbeláez, P., Girshick, R., & Malik, J. (2014). Simultaneous detection and segmentation. In *Computer vision–eccv 2014* (pp. 297–312). Springer.

Hart, N. S. (2001a). Variations in cone photoreceptor abundance and the visual ecology of birds. *Journal of Comparative Physiology A*, *187*(9), 685–697.

Hart, N. S. (2001b). The visual ecology of avian photoreceptors. *Progress in retinal and eye research*, *20*(5), 675–703.

Hart, N. S., & Hunt, D. M. (2007). Avian visual pigments: characteristics, spectral tuning, and evolution. *the american naturalist*, *169*(S1), S7–S26.

Hough, P. V. (1962). *Method and means for recognizing complex patterns* (Tech. Rep.).

Hubel, D. H., & Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, *195*(1), 215–243.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., ... Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.

Karayev, S., Trentacoste, M., Han, H., Agarwala, A., Darrell, T., Hertzmann, A., & Winnemoeller, H. (2013). Recognizing image style. *arXiv preprint arXiv:1311.3715*.

Kavallieratou, E., Sgarbas, K., Fakotakis, N., & Kokkinakis, G. (2003). Handwritten word recognition based on structural characteristics and lexical support. In *Document analysis and recognition, 2003. proceedings. seventh international conference on* (pp. 562–566).

Knudson, A. G. (1971). Mutation and cancer: statistical study of retinoblastoma. *Proceedings of the National Academy of Sciences*, *68*(4), 820–823.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, *3361*(10), 1995.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, *1*(4), 541–551.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

LeCun, Y., Kavukcuoglu, K., Farabet, C., et al. (2010). Convolutional networks and applications in vision. In *Iscas* (pp. 253–256).

Levi, G., & Hassner, T. (2015a). Age and gender classification using convolutional neural networks. In *Proceedings of the ieee conference on computer vision and pattern recognition workshops* (pp. 34–42).

Levi, G., & Hassner, T. (2015b). Emotion recognition in the wild via convolutional neural networks and mapped binary patterns. In *Proceedings of the 2015 acm on international conference on multimodal interaction* (pp. 503–510).

Lin, M., Chen, Q., & Yan, S. (2013). Network in network. *arXiv preprint arXiv:1312.4400*.

Long, J., Shelhamer, E., & Darrell, T. (2014). Fully convolutional networks for semantic segmentation. *CoRR*, *abs/1411.4038*. Retrieved from `http://arxiv.org/abs/1411.4038`

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 3431–3440).

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. the proceedings of the seventh ieee international conference on* (Vol. 2, pp. 1150–1157).

Lupaşcu, C. A., & Tegolo, D. (2010). Automatic unsupervised segmentation of retinal vessels using self-organizing maps and k-means clustering. In *Computational intelligence methods for bioinformatics and biostatistics* (pp. 263–274). Springer.

Pahalawatta, K. K., & Green, R. (2012). Detection and classification of nano-scale particles with image histograms. In *Proceedings of the 27th conference on image and vision computing new zealand* (pp. 446–451).

Pradhan, R., Kumar, S., Agarwal, R., Pradhan, M. P., & Ghose, M. (2010). Contour line tracing algorithm for digital topographic maps. *International Journal of Image Processing (IJIP)*, *4*(2), 156–163.

Ram, K., & Sivaswamy, J. (2009). Multi-space clustering for segmentation of exudates in retinal color photographs. In *Proc. annu. int. conf. ieee eng. med. biol. soc* (pp. 1437–1440).

Rodieck, R. (2003). The density recovery profile: a method for analysis of points in the plane applicable to retinal studies. *Visual Neuroscience*, *20*(03), 349–349.

Rohatgi, V. K. (2003). *Statistical inference*. Courier Corporation.

Ross, A., & Shah, S. (2006). Segmenting non-ideal irises using geodesic active contours. In *Biometric consortium conference, 2006 biometrics symposium: Special session on research at the* (pp. 1–6).

Rouhi, R., Jafari, M., Kasaei, S., & Keshavarzian, P. (2015). Benign and malignant breast tumors classification based on region growing and cnn segmentation. *Expert Systems with Applications*, *42*(3), 990–1002.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

Shaoqing Ren, R. G. J. S., Kaiming He. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sivic, J., Russell, B. C., Efros, A. A., Zisserman, A., & Freeman, W. T. (2005). Discovering objects and their location in images. In *Computer vision, 2005. iccv 2005. tenth ieee international conference on* (Vol. 1, pp. 370–377).

Suga, A., Fukuda, K., Takiguchi, T., & Ariki, Y. (2008). Object recognition and segmentation using sift and graph cuts. In *Pattern recognition, 2008. icpr 2008. 19th international conference on* (pp. 1–4).

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A. (2014). Going deeper with convolutions. *CoRR*, *abs/1409.4842*. Retrieved from `http://arxiv.org/abs/1409.4842`

Tang, M. (2011). Recognizing hand gestures with microsofts kinect. *Palo Alto: Department of Electrical Engineering of Stanford University:[sn]*.

Tian, Q.-C., Pan, Q., Cheng, Y.-M., & Gao, Q.-X. (2004). Fast algorithm and application of hough transform in iris segmentation. In *Machine learning and cybernetics, 2004. proceedings of 2004 international conference on* (Vol. 7, pp. 3977–3980).

Uijlings, J. R., van de Sande, K. E., Gevers, T., & Smeulders, A. W. (2013). Selective search for object recognition. *International journal of computer vision*, *104*(2), 154–171.

Vlahou, A., Schorge, J. O., Gregory, B. W., & Coleman, R. L. (2003). Diagnosis of ovarian cancer using decision tree classification of mass spectral data. *BioMed Research International*, *2003*(5), 308–314.

Vorobyev, M. (2003). Coloured oil droplets enhance colour discrimination. *Proceedings of the Royal Society of London B: Biological Sciences*, *270*(1521), 1255–1261.

Wei, Y., Xia, W., Huang, J., Ni, B., Dong, J., Zhao, Y., & Yan, S. (2014). CNN: single-label to multi-label. *CoRR*, *abs/1406.5726*. Retrieved from `http://arxiv.org/abs/1406.5726`

Wu, Y. C., Freedman, M. T., Hasegawa, A., Zuurbier, R. A., Lo, S.-C. B., & Mun, S. K. (1995). Classification of microcalcifications in radiographs of pathologic specimens for the diagnosis of breast cancer. *Academic Radiology*, *2*(3), 199–204.

Xu, J., Schwing, A., & Urtasun, R. (2014). Tell me what you see and i will show you where it is. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 3190–3197).

Yuen, H., Princen, J., Illingworth, J., & Kittler, J. (1990). Comparative study of hough transform methods for circle finding. *Image and vision computing*, *8*(1), 71–77.

Zhang, N., Donahue, J., Girshick, R., & Darrell, T. (2014). Part-based r-cnns for fine-grained category detection. In *Computer vision–eccv 2014* (pp. 834–849). Springer.

Zhang, Y., Zhaoxing, Z., & Han, X. (2009). Category specific sift descriptor and its combination with color information for content-based image retrieval. In *Proceedings of the 2nd international conference on interaction sciences: Information technology, culture and human* (pp. 685–690).

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., & Oliva, A. (2014). Learning deep features for scene recognition using places database. In *Advances in neural information processing systems* (pp. 487–495).

Zitnick, C. L., & Dollár, P. (2014). Edge boxes: Locating object proposals from edges. In *Computer vision–eccv 2014* (pp. 391–405). Springer.