4-27-2022

Eye-Tracking using Deep Learning

Sam Trenter
University of Missouri-St. Louis, sntrenter@gmail.com

Eye-Tracking using Deep Learning

by

Samuel Trenter

A Thesis
Submitted to The Graduate School of the

University of Missouri-St. Louis
in partial fulfillment of the requirements for the degree

Master of Science

In

Computer Science

April 2022

Advisory Committee

Badri Adhikari, Ph.D.
(Chairperson)

Sharlee Climer, Ph.D.

Mark Hauschild, Ph.D.

Abstract

Eye-tracking can be valuable for researchers in many domains. Most eye-tracking technologies require an extra piece of costly hardware. Several other available eye-tracking solutions are usually not very accurate and require a costly subscription. Our project was oriented at creating a free and open-source alternative that does not require additional equipment. We developed a deep learning-based solution as a prototype for this project. Specifically, we developed a deep learning model to predict a user's gaze position on the screen. We created our training data set using a commercially available eye-tracker to train the model. Each training sample consists of a video frame of a person looking at the screen (input feature) and the corresponding true gaze location on the screen (output labels). When using a model with a gaze location output on a 2d plane, a deep learning model was able to in most cases interpret and predict a correct gaze position. Our results demonstrate that with minimal processing a deep learning algorithm can be effectively used to create and deploy eye tracking solutions.

Contents

List of Figures

Chapter 1

Introduction

1.1 Current technologies

Eye tracking is the process in which someones eye movements are studied. This most commonly done by tracking pupil with near infrared video. The devices output infrared light and calculate gaze position using the difference between the center of the pupil and the center of the light source. Both gaze position and pupil tracking are very helpful tools in many fields including psychology, marketing, and user interface design.

What is very exciting is that recently the use of eye tracking in the field of computer science has grown greatly in the last few years. This is good but the methods of eye tracking described in these papers are not what is traditionally considered eye tracking. The first paper uses eye tracking to generate segmentation masks that will be sent to a machine learning model later. This method uses a traditional infrared eye tracker but shows that Eye tracking can be used to generate data for machine learning purposes. [22] The next paper also uses eye tracking, and gathers data in a similar way we gathered eye data. but uses it for Eye state recognition with the purpose of getting a driver drowsiness state. They do this by sending a deep learning model pictures of the eyes to get predictions on whether the eye is in an open or closed state. [24] Other papers have researched whether eye tracking can help predict future moves in games like chess. [15] This shows that there is an interest in eye tracking, but for how useful the tool has the potential to be there is a surprising lack of access.[11] Some researchers have made attempts to make eye tracking more open to the public but those efforts haven't resulted in mainstream adoption and were only available on IOS devices. [13]

1.2 Limitations

Most limitations with eye tracking fall into two categories, first being cost. While there are some cheap eye trackers, many of the low cost options don't allow for development use with standard development kits. This makes it extremely hard to do any eye tracking research because you have to make all your own tools. That leaves you with research or commercial grade models, these can vary greatly in price and sometimes can only be rented instead of bought outright. It is also hard to get an estimate for eye trackers because most commercial and research grade eye trackers must be bought direct and usually require talking with a sales Representative. [11]

The unwillingness to list prices also brings up my second category, companies in the eye tracking space don't only sell to researchers, they sell to advertising and marketing firms. Because there is a large commercial market to eye tracking technology, scientific research sometimes takes a backseat to the marketing and advertising industries. This means it can be difficult to access any eye tracking technology either from it being proprietary industry technology or from it being stuck behind a paywall. There are some companies and research teams that are working to create solutions to this issue. [11] [13] but the majority of commercial products can be hard to obtain for the average researcher.

1.3 Goals

The intent of this research is to find out if the two limitations mentioned above can be removed by using deep learning to create a simple eye tracker. Our aim was not to completely mirror what current eye trackers could do but to see if it was possible to make tools that could be used by teams who could not acquire eye trackers but did have access to a webcam. Finally, after reviewing the results, we will suggest future research to be done with this topic.

Chapter 2

Background and related work

Deep Learning, Eye Tracking, and Health research have Recently started to become more intertwined as the projects involving them slowly start to over lap. What is unique about these three research topics is how it can uniquely contribute to each other. For deep learning it could provide us with possible new input methods like eye tracking data or gaze heat maps from eye tracker. For eye tracking this research could provide an easier way for people to begin to collect eye tracking data. We also see more and more medical data, usually in the form of scans and images, as inputs for models. For Health research, having new tools to use and possibly even new devices to test patients could provide vital information about a patients health.

2.1 How health research uses eye tracking

There are a few data points we are interested when using eye tracking. Saccades are fast and accurate eye movments used to take in a new enviroment. [21] Next are fixations, this is when they eyes will focus on a singal object to take in information [21]. There is smooth pursuit, which is when tracking a moving target. As aposed to saccades, these are mostly smooth movements with minimal jumping. This usually only happens when observing slowely moving targets [21].

Some of the data mentioned above can actually be used to predict medical issues, like using saccades to predict problems steming from alzheimers disease [20].Heat maps have been used in coordination with lip movment to help visualize mental health [9]. Research has even been done to study the physiological response between subjects when presented with different landscapes[14]. Fixations have actually been used to study the ability of warning labesl to attract the attention of users [8]. While this new intake of research with eye tracking is exciting there is still much to learn in regards to how to properly do eye tracking studies. Researchers need to be careful and make sure they are using the best methods avalible when collecting data [3].

What is extremely interesting is the variety of applications that eye tracking has been able to be used for. The first use of actual eye trackers was in 1947. It was used by the U.S. Air Force to help determine an optimal placement for instrument controls for a plane [21]. It has also been used in conjunction with medical forms to help provide relevant medical information that a doctor may be looking for [10]. One area where eye tracking may become the most useful is in medical imaging. Eye tracking has found that experienced doctors more efficiently search through scans to find critical features. This could provide a valuable metric when training new doctors and help give valuable insight into how good someone is at reading a medical scan [2].

2.2 How health research uses deep learning

Deep learning is increasingly becoming a Swiss army knife of the artificial intelligence field with it's ability to accomplish many different tasks. as Francois Challet puts it, "the only real success of deep learning so far has been the ability to map space X to space Y using a continuous geometric transform, given large amounts of human-annotated data. Doing this well is a game-changer for essentially every industry, but it is still a very long way from human-level AI." [4] In his essay, "The limitations of deep learning" he mentions that deep learning models don't have to understand any of their input in a human way, but by using large data sets deep learning models are able to train them to map data to human concepts on specific problems [4]. While this may seem limiting, it allows researches to do many things that would be considered unimaginable a decade ago.

One of the more relevant areas where deep learning overlaps with medical research is imaging. One of the hardest parts about training doctors is teaching them effective methods "reading" medical images or scans [12]. On top of that, the data that is getting generated by the field of bio imaging is increasing at alarming rates. To keep up with the new data being generated, new methods and practices of data handling and parallelization must be used to fully realize the potential of the data available[18]. To make use of all of the data being generated, new methods must be created and deep learning may provide a good way of orginizing and sorting large amount of data with minimal human intervention. Infact success has already been found in this area by using a deep learning model to sort Retinal photographs [17]. This shows that while we have started using deep learning, we still have a lot of potential left.

Deep learning has shown to be an extremely useful tool in many fields but one thing that is required for deep learning models to be successful is large labeled data sets [4]. While many data sets may exist, to completely take advantage of deep learning in the medical field work must be done to create new data sets.

An example of a deep learning model helping with data segmentation is using a model to process a large amount of images or scans with a similar accuracy to what a human could achieve. Thanks to the ability to multi process work, hundreds of scans could be looked at by a deep learning model in the time it takes for a human to look at one scan [7]. While it doesn't replace the ability of a well trained experienced physician, a deep learning model is able to greatly assist in a physicians ability to accurately read scans or images [7]. It may be a long time before we are even close to replacing the observations of a physician when viewing medical scans but more and more areas of medicine are using deep learning models not to predict whether or not a scan has a tumor or something serious, but as a method to highlight areas of concern for observation. An example would be using a deep learning algorithm to find areas of a mammogram that appear to contain a tumor[16]. These show that while not completely replacing a doctor or physician, a deep learning model may add a very capable tool to their toolbelt.

2.3 How Health Research, Deep Learning, and Eye Tracking Work Together

We've seen how crucial these technologies can be to their respective fields but if combined they have the opportunity to create great positive change. For example, a deep learning model was able to learn to predict tumor locations using both and audio recording and gaze data of brain scans. Not only is this research taking advantage of eye data but also speech data recorded by a physician going over a scan [23]. Research doesn't need to be that complicated though. Research has been done to make use of eye tracking for creating data sets for deep learning networks. Like mentioned previously, Deep learning networks require a large data set to make accurate predictions [4]. It is however possible to use eye tracking to aid in the creation of data sets for deep learning. A study in 2019 showed that eye tracking can be used to create segmentation masks much faster than traditional point and click methods. This saves valuable researcher time, and allows for the creation of larger data sets faster[22]. These works show us that there is a large amount of potential when it comes to research around these topics and that there is a lot left to learn.

Chapter 3

Methods

3.1 Data set creation

Our Data set what created using a Camera recording in 640x480 at 30 frames per second and an eye tracking recording 30 Gaze points per second. Every Frame of the video and gaze point was recorded with a time stamp and them combined after. Having an image and a gaze point pair then allows us to form a data set in a way that is more easily readable by a model. Images would be separated based on what gaze location they were associated with. Finally We have a script parse through the data set and reduce all images to the area just around the eyes. The script will also take images that might not have both eyes in frame or has the viewer looking off screen and remove them. We found that reducing the image from a native webcam picture the just the area around the eyes greatly improved the accuracy of the model and seemed to eliminate most issues with noise entering the model.

3.2 Optimal screen representation

A large problem throughout this project was how to represent the screen the viewer was looking at. A typical computer screen is 1920x1080 pixels and the app used to record screen position was 1920x900 pixels. If using a 1 pixel per class, those size of screens would leave you with 2 million classes to deal with. Fortunately it is not only possible to but easy to use a regression based method for this data set.

This does help a with many of our issue but it does not stop the issue of data sparsity. With the around 2 million possible pixel locations, we would need to record continuously while only looking at new pixels for 19.6 hours to get a ”complete” data set. That is one a data set only one image per pixel. Our data set has a slight bias to the bottom left of the screen causing it to predict that area more than baseline.

3.3 Model

The model we ended up using for this is a pretty standard Deep learning model. Two things that ended up being consistently helpful for our usage was a reduced learning rate and using padding. While padding being helpful is not surprising the learning rate being lower gives us some insight into what a model might need to be tuned for eye-tracking. It seems that a model needs to make much smaller changes to adjust to picking up the eyes from an image. this is not very surprising considering how small the area the model wants to look at is. Without the lower learning rate the model had a tendency to increase its loss instead of decreasing it's loss.

```
model = Sequential()
model.add(Conv2D(filters =32, kernel_size = (3,3), activation = '
                                  relu', input_shape = (40, 100, 3)
                                  ,padding = 'same'))
model.add(Activation('relu'))
model.add(Conv2D(32, (3, 3),padding = 'same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(16, (3, 3),padding = 'same'))
model.add(Activation('relu'))
model.add(Conv2D(8, (3, 3),padding = 'same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(4, (3, 3),padding = 'same'))
model.add(Activation('relu'))
model.add(Flatten())
model.add(Dense(32 * 32))
model.add(Activation('relu'))
opt = keras.optimizers.Adam(learning_rate=.0001)
model.compile(optimizer = opt, loss = 'mean_squared_error', metrics
                                   = ['mean_absolute_error'])
print(model.summary())
```

3.4 Model input

A major difficulty in the project was choosing how to format the data, and which method of formatting would make it easiest for the model to learn. We started with sending it 640x480 PNG files, but found that the model preformed very poorly on these. To work through this we used OpenCV to only send the model a 100x40 picture of the the eyes.

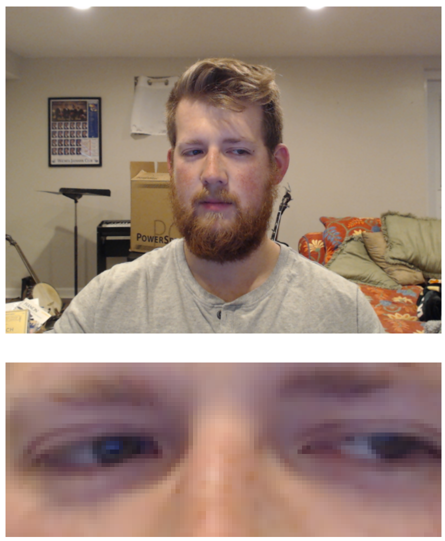

Figure 3.1: Above is an example of what an original image would look like and below is what would get sent to the model.

Adjusting the image to just the eyes immediately yielded better results. This also added some processing time but not too much to disallow using this model live. To accomplish the eye cropping we created an algorithm to detect a facial cascade using openCV, then if a face was detected look for two eyes in the upper half of the face region. If two could be found it would save the image into a new data set. While this method does remove some images from the data set we found it to be an acceptably small portion.

One of the largest issues we encountered while researching this problem was the data sparsity problem. Currently the average screen is 1920x1080 pixels. Current eye trackers output a pixel value for gaze position between 0 and 1920 for width and 0 and 1080 for height. If we wanted to create a multi class problem of that size it would need to have an output row with a length of two million.

To combat this we chose to decrease to resolution of our output as it relates to our screen resolution. So our 32x32 output is a transformed version of what would be our preferred 1920x1080 output. We think this is appropriate because while many modern eye trackers do give a discrete pixel value, they are rarely have pixel perfect accuracy. While a 32x32 output doesn't give enough resolution to make out fine details in a gaze it can give a general idea as to where someone was looking on screen and provide valuable data. With the 1920x1080 screen reduced to 32x32, each pixel in the 32x32 grid corresponds to a 60x28 pixel location on the 1920x1080 grid.

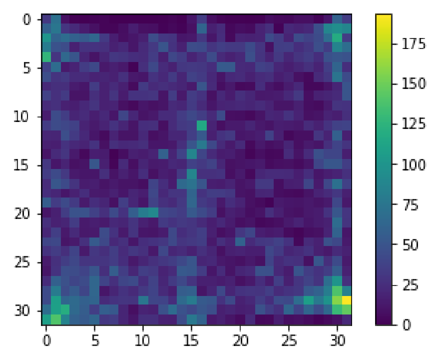

Figure 3.2: Heat map of data set as 32x32 grid

Above is our data set arranged in a 32x32 grid. this is what we went with in the final implementation. We found that 64x64 also showed promise but results were not as clear. Smaller grids were also experimented with but we found that they were trivial for the model.

Above is our model arranged in a 960x450 grid. this is about half the size of the 1920x1080 screen. As you can see here the sparsity of the data set would make it almost impossible for any model to make meaningful predictions with so many empty pixel locations.

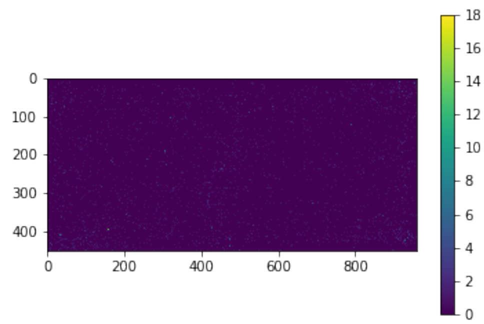

Figure 3.3: Heat map of data set as 960x450 grid

Chapter 4

Output

Throughout experimentation with this project, many different output methods were tried, with the two main being A class based model and a regression based model. Both model outputs work well, but the regression based model was able to learn much more than the class based model.

Both of them use an output formatted to a 32x32 grid, which represents the gaze position corresponding to the input image. The Class based model has one correct prediction while the Regression based model uses a Gaussian distribution to reward answers that are spatially close.

4.1 Multi-Class classification

For the majority of this research a class based model was used for predicting gaze position. The problem with using a class based model architecture is that the model has no spacial awareness in regards to wrong answers. Meaning that a gaze position that is 1 pixel away is treated the same as a pixel location that is 32 pixels away.

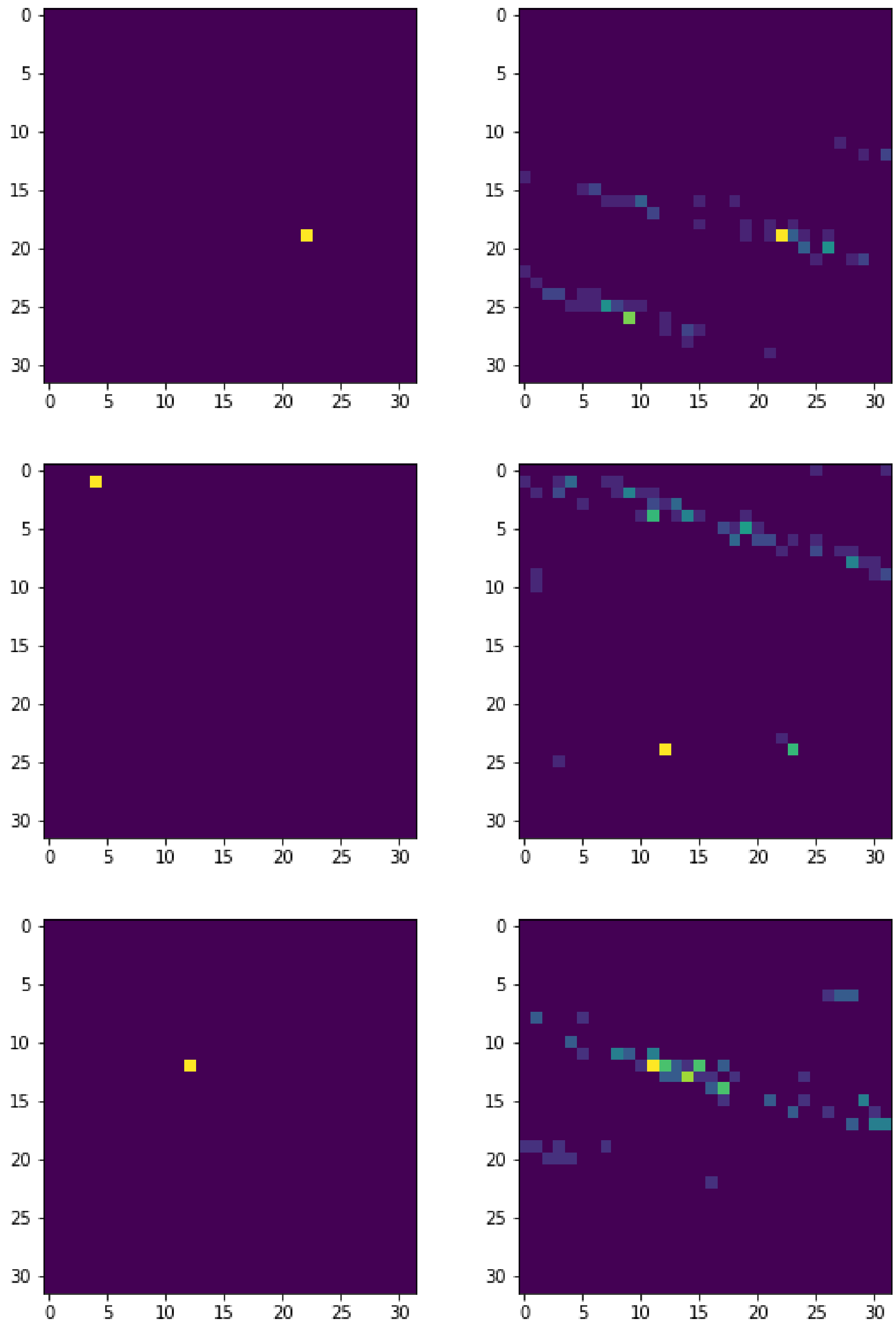

Figure 4.1: A few examples of predicted images using the class based model.

The model can learn similarly to how the regression based model does, but is limited extremely early by it's inability to spatially relate pixel locations. Interestingly the model would output diagonal lines in an area it was trying to predict. It does not appear to be caused by the data distribution, if it did we would see it start to match the heat map similarly to how the regression based model does. I Think the diagonal lines show us that this version of the model especially struggles to differentiate between images on a diagonal.

4.2 Regression based model

Using continuous values in a 32x32 grid for our models output allowed for use to customize how specific our model needed to be. We used a simple Gaussian distribution where the gaze location was located. This allowed the model to receive "partial credit" when getting close to the correct gaze location. The size of the Gaussian distribution was difficult to fine tune because too small would lead the model to make no predictions. A Gaussian distribution too large would lead to a large blob in the center of the 32x32 grid being predicted for every input.

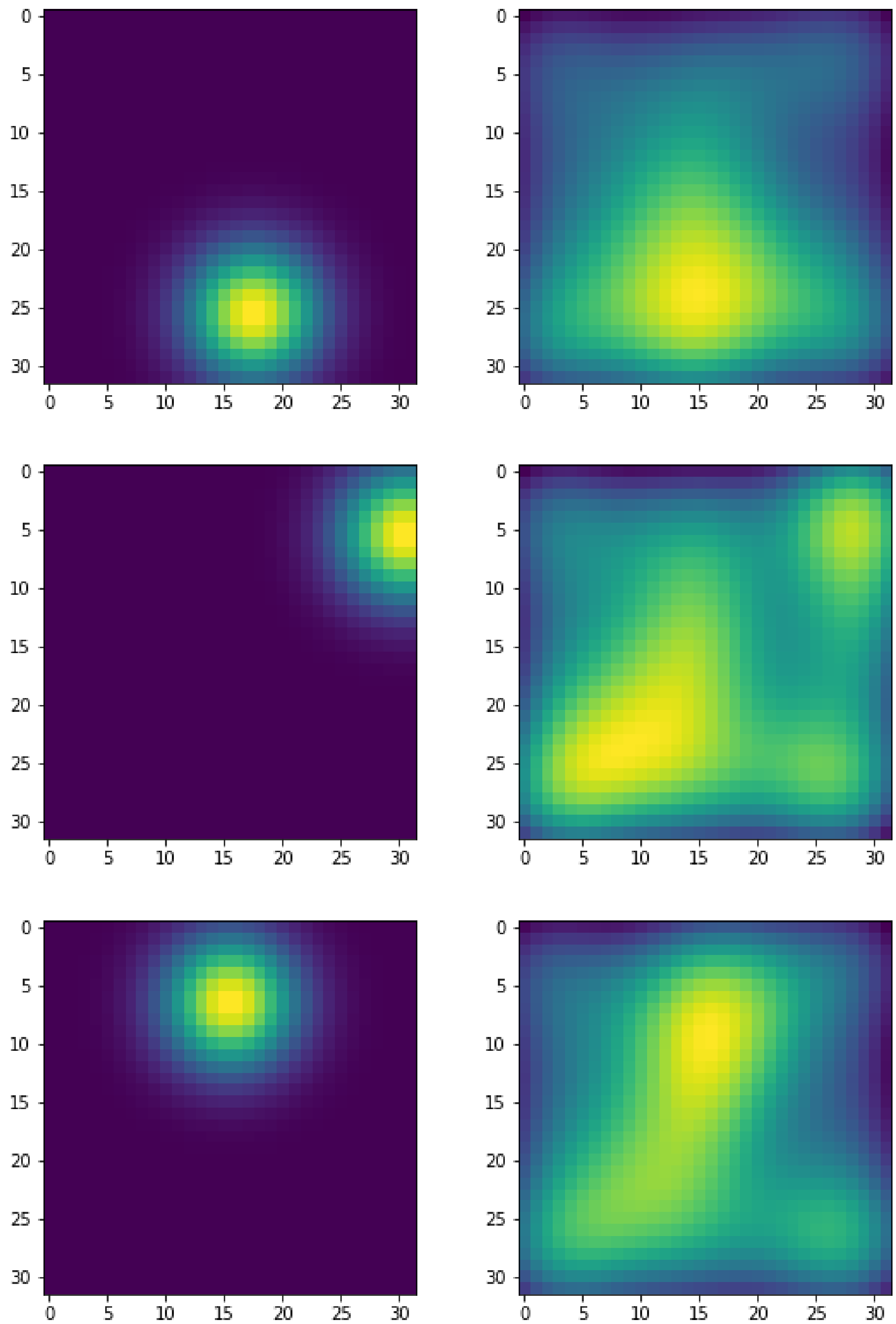

Figure 4.2: A few examples of predictions with the regression based model. Images on the left are Y and images on the right are the models prediction.

If we only take into account its max value as a prediction, The regression based model preforms much better than the class based model. Compared to the class based model, the regression based model seems to show much more consistency when show a series of images. This leads me to believe that it was learning more from the data it had available compared to what the class based model was able to learn.

One thing this model does have a problem with is data set bias. You can note in the three predictions shown all have a blob highlighted in the bottom left and a section in the bottom right highlighted. If you look at the heat map of the data set you will notice that there is a large segment in the bottom left that has slightly more images per pixel on average than the rest of the data set. You will also note the corresponding bright yellow spot where many images correspond to one pixel. You can see that in the second set of images the bias out weights what would be the true value, this would lead to a incorrect gaze prediction. These sections in the data set appear to just be random artifacts from recording the data set.

4.3 Results

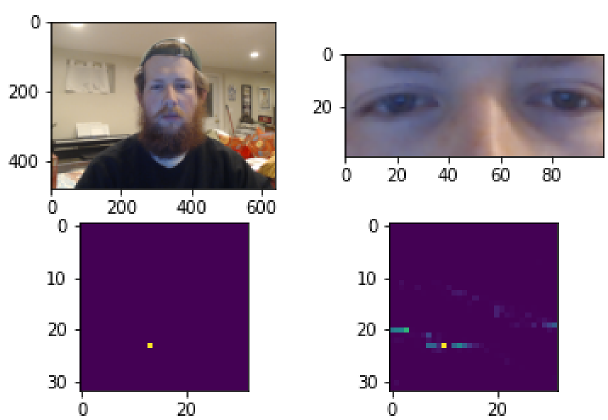

Figure 4.3: This is a prediction using our class based model. While it does preform fairly well in this case results like this were not consistent.

This really shows the limitations of the class based model. I think while limited they show that even with limited spacial knowledge, as in the model was not able to get loss information from how far away and in what direction the predictions were off it shows that a model can begin the make fairly reasonable predictions while being fed very limited information and feedback. I think there

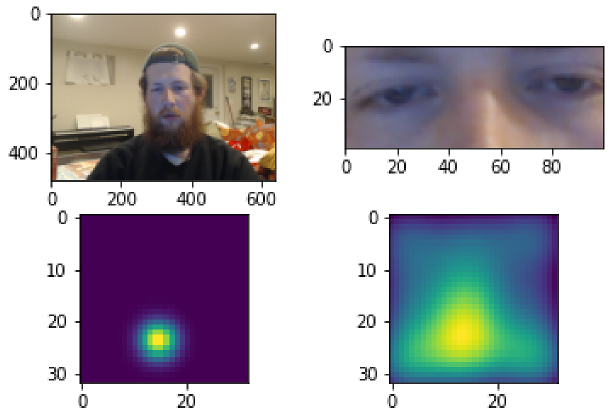

Figure 4.4: Here is an example of when our prediction was correct, but you can see a large section of the 32x32 output still highlighted

isn't much potential with future research for class based model outputs. While class based models were great for debugging early models that had a very small output space, think 2x1 or 2x2, they limit your potential learning for larger models greatly.

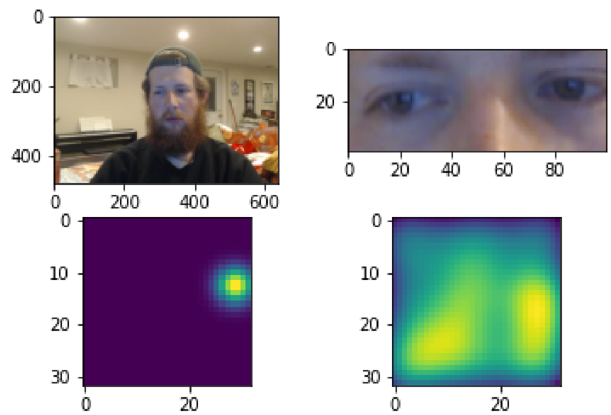

Figure 4.5: Here we see our prediction was correct, because the highest highlighted section corresponds to the correct region of the expected output but we still see a large section that the model predicts confidently, this is because of the data set bias we have.

These figures show that an image can be taken from a webcam and with slight alteration can be fed to a neural network, and that network can output a gaze position. These results also show that using the model to predict gaze position instead of pupil position could be to large a task. They also show that a multi output regression is most likely the type of output to use moving forward.

It is not very surprising that we had such major improvements with models using the multi-output regression. In other works that have done eye tracking in a similar manner, great success has been seen. although those were with mobile devices held much closer to the face and much larger data sets [13]. What we can see clearly demonstrated here is that eye tracking can be done with a web cam, but like with previous research a large diverse data set will be needed to gain reasonable accuracy for live use.

Chapter 5

Conclusions

Deep learning can be used to predict gaze position. Simple image processing makes it much easier for model to pick up on eye movement and predict gaze-position. The hardest part about training a model was the sparsity of a data set, This sparsity makes it especially hard to create and store data sets that could lead to creating a more generalized model. With improvements a model should be fast enough to make live gaze-position predictions.

5.1 Future work

It tasked with redoing this with the knowledge I have now I would begin by training a model to predict pupil position instead of gaze-position, while this might be harder to initially gather a data set and to train, it should be possible to take pupil position and use it to calculate gaze-position. This method would be more similar to how current eye trackers work and should work better. I would also spend more time researching how to mock data for to feed the model. Seeing the success other teams had with much larger data set I think the future of eye tracking will involve generating participant data. Researching sending inputs like "face direction" and "eye location" relative to the camera might also yield interesting results.

Bibliography

[1] Robert G. Alexander et al. "What do radiologists look for? Advances and limitations of perceptual learning in radiologic search". In: *Journal of Vision* 20.10 (Oct. 2020), pp. 17–17. ISSN: 1534-7362. DOI: 10.1167/jov.20.10.17. eprint: https://arvojournals.org/arvo/content_public/journal/jov/938495/i1534-7362-20-10-17_1602674199.67748.pdf. URL: https://doi.org/10.1167/jov.20.10.17.

[2] Tad T. Brunyé et al. *Eye tracking reveals expertise-related differences in the time-course of medical image inspection and diagnosis*. 2020. URL: https://doi.org/10.1117/1.JMI.7.5.051203.

[3] Benjamin T. Carter and Steven G. Luke. "Best practices in eye tracking research". In: *International Journal of Psychophysiology* 155 (2020), pp. 49–62. ISSN: 0167-8760. DOI: https://doi.org/10.1016/j.ijpsycho.2020.05.010. URL: https://www.sciencedirect.com/science/article/pii/S0167876020301458.

[4] Francois Chollet. "The limitations of deep learning". In: *Deep learning with Python* (2017).

[5] Federica Cilia et al. "Computer-Aided Screening of Autism Spectrum Disorder: Eye-Tracking Study Using Data Visualization and Deep Learning". In: *JMIR Hum Factors* 8.4 (Oct. 2021), e27706. ISSN: 2292-9495. DOI: 10.2196/27706. URL: http://www.ncbi.nlm.nih.gov/pubmed/34694238.

[6] Kirsten A. Dalrymple et al. "Machine learning accurately classifies age of toddlers based on eye tracking". In: *Scientific Reports* 9.1 (2019). DOI: 10.1038/s41598-019-42764-z.

[7] Mohammad Hesam Hesamian et al. "Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges". In: *Journal of Digital Imaging* 32.4 (Aug. 2019), pp. 582–596. ISSN: 1618-727X. DOI: 10.1007/s10278-019-00227-x. URL: https://doi.org/10.1007/s10278-019-00227-x.

[8] Ji-Eun Hwang et al. *Differences in visual fixation duration according to the position of Graphic Health Warning Labels: An eye-tracking approach*. Sept. 2018. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6659495/.

[9] Emi Katada et al. “Visualization and Meanings of Mental Health State Based on Mutual Interactions Between Eye-gaze and Lip Movements”. In: *2018 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*. 2018, pp. 18–23. DOI: 10.1109/ISPACS.2018.8923429.

[10] Andrew J King et al. “Leveraging Eye Tracking to Prioritize Relevant Medical Record Data: Comparative Machine Learning Study”. In: *J Med Internet Res* 22.4 (Apr. 2020), e15876. ISSN: 1438-8871. DOI: 10.2196/15876. URL: http://www.ncbi.nlm.nih.gov/pubmed/32238342.

[11] Author:nbsp;Anna Kirkpatrick. *Low cost eye tracking for computer access and speech synthesis*. URL: https://www.washington.edu/accesscomputing/resources/accesscomputing-news-septembe018/low-cost-eye-tracking-computer-access-and-speech-synthesis.

[12] Ellen M. Kok et al. “What We Do and Do Not Know about Teaching Medical Image Interpretation”. In: *Frontiers in Psychology* 8 (2017). ISSN: 1664-1078. DOI: 10.3389/fpsyg.2017.00309. URL: https://www.frontiersin.org/article/10.3389/fpsyg.2017.00309.

[13] Kyle Krafka et al. 2016. URL: https://gazecapture.csail.mit.edu/.

[14] Congcong Liu et al. *Viewing garden scenes: Interaction between gaze behavior and physiological responses*. May 2020. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7881881/.

[15] Justin Le Louedec et al. “Deep Learning Investigation for Chess Player Attention Prediction Using Eye-Tracking and Game Data”. In: ETRA ’19 (2019). DOI: 10.1145/3314111.3319827. URL: https://doi.org/10.1145/3314111.3319827.

[16] Dzung L. Pham, Chenyang Xu, and Jerry L. Prince. “Current Methods in Medical Image Segmentation”. In: *Annual Review of Biomedical Engineering* 2.1 (2000). PMID: 11701515, pp. 315–337. DOI: 10.1146/annurev.bioeng.2.1.315. eprint: https://doi.org/10.1146/annurev.bioeng.2.1.315. URL: https://doi.org/10.1146/annurev.bioeng.2.1.315.

[17] Tyler Rim et al. “Deep Learning for Automated Sorting of Retinal Photographs”. In: *Ophthalmology Retina* 4 (Mar. 2020). DOI: 10.1016/j.oret.2020.03.007.

[18] Ingrid Scholl et al. “Challenges of medical image processing”. In: *Computer Science - Research and Development* 26.1 (Feb. 2011), pp. 5–13. ISSN: 1865-2042. DOI: 10.1007/s00450-010-0146-9. URL: https://doi.org/10.1007/s00450-010-0146-9.

[19] N. Selvi and C. Lakshmi. “A review: Towards quality improvement in real time eye-tracking and gaze”. In: *International Journal of Applied Engineering Research* 10 (Jan. 2015), pp. 15731–15746.

[20] Ruxsana Shafiq-Antonacci et al. "Spectrum of Saccade System Function in Alzheimer Disease". In: *Archives of Neurology* 60.9 (Sept. 2003), pp. 1272–1278. ISSN: 0003-9942. DOI: 10.1001/archneur.60.9.1272. eprint: https://jamanetwork.com/journals/jamaneurology/articlepdf/784658/noc20013.pdf. URL: https://doi.org/10.1001/archneur.60.9.1272.

[21] Hari Singh. "Human Eye Tracking and Related Issues: A Review". In: *International Journal of Scientific and Research Publications* 2 (Sept. 2012), pp. 1–9.

[22] J. N. Stember et al. *Eye tracking for deep learning segmentation using Convolutional Neural Networks - Journal of Digital Imaging.* May 2019. URL: https://link.springer.com/article/10.1007/s10278-019-00220-4#citeas.

[23] Joseph N. Stember et al. "Integrating Eye Tracking and Speech Recognition Accurately Annotates MR Brain Images for Deep Learning: Proof of Principle". In: *Radiology: Artificial Intelligence* 3.1 (2021). PMID: 33842890, e200047. DOI: 10.1148/ryai.2020200047. eprint: https://doi.org/10.1148/ryai.2020200047. URL: https://doi.org/10.1148/ryai.2020200047.

[24] Lei Zhao et al. "Eye state recognition based on Deep Integrated Neural Network and Transfer Learning". In: *Multimedia Tools and Applications* 77.15 (2017), pp. 19415–19438. DOI: 10.1007/s11042-017-5380-8.