2018-04-23

Automatic Eye-Gaze Following from 2-D Static Images: Application to Classroom Observation Video Analysis

Arkar Min Aung
Worcester Polytechnic Institute

Automatic Eye-Gaze Following from 2-D Static Images: Application to Classroom Observation Video Analysis

by

Arkar Min Aung

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Computer Science

by

April 2018

APPROVED:

Professor Jacob R. Whitehill, Major Thesis Advisor

Professor Neil T. Heffernan, Thesis Reader

Professor Craig E. Wills, Head of Department

Abstract

In this work, we develop an end-to-end neural network-based computer vision system to automatically identify where each person within a 2-D image of a school classroom is looking ("gaze following"), as well as who she/he is looking at. Automatic gaze following could help facilitate data-mining of large datasets of classroom observation videos that are collected routinely in schools around the world in order to understand social interactions between teachers and students. Our network is based on the architecture by Recasens, et al. (2015) but is extended to (1) predict not only where, but who the person is looking at; and (2) predict whether each person is looking at a target inside or outside the image. Since our focus is on classroom observation videos, we collect gaze dataset (48,907 gaze annotations over 2,263 classroom images) for students and teachers in classrooms. Results of our experiments indicate that the proposed neural network can estimate the gaze target - either the spatial location or the face of a person - with substantially higher accuracy compared to several baselines.

Acknowledgements

I would like to express my deepest gratitude to my advisor, Professor Jacob Whitehill, for his professional mentoring, patience, attention to details and close collaboration during this research. I have been extremely lucky to have an advisor who not only cared about my work but also responded to my queries so promptly. Working on my Master's Thesis was one of the most challenging but rewarding experiences, and I am honored to work with Professor Whitehill.

I also want to thank my thesis reader and academic advisor, Prof. Neil Heffernan, for all the great advice and directions he has given me throughout my Master's degree. I also would like to thank Professor Heffernan for his feedback on this thesis work during Educational Data Mining Research Group meetings.

I would like to thank my father, U Min Aung, my mother, Daw Shereen Ahad and my sister, Darli Min Aung. Due to their unwavering support throughout my roughest patches during my graduate studies, I could pull myself and keep on walking. Moreover, I would like to thank Ma Shwe Sin Oo, for all for her love and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

When a person suddenly looks outside the room, others in the group suddenly follow the gaze of that person to assess what is drawing that person's attention. What a person is looking at can be a good indicator of what is on that person's mind. Therefore, gaze of a person can tell more than just where they are looking at. Gaze of a person can be useful for inferring the attention, thoughts and intentions of a person [3, 54]. Gaze of a person is regarded "as a perceptual/psychological act through which other people can glean information about the gazer's perceptions, desires, emotions, and intentions" [11].

Advances in novel machine learning techniques, abundance of data and computing power have given machines the capacity to become more integrated in society (e.g. autonomous transportation, health care, security, and education). However, this means that machines need to have better understanding of humans' intentions. For example, a self-driving car has to be aware of the attention and intentions of a pedestrian, so that it can perform more informed path planning and driving behavior adjustments. If a self-driving car has a computer vision system which can infer the intentions and the object of attention of a pedestrian from his or her gaze direction, better adjustments can be made compared to systems which are not aware of a pedestrian's attention.

Not only will machines which can follow people's gaze direction help build future vehicles, but they can also help build smart technologies that can be integrated in other environments. In a classroom context, disengaged students (e.g. gazing outside) or students in need of help (e.g. not knowing what to do and repeatedly looking at others' work) can be identified by recognizing where they are looking at. Since a single classroom may have multiple students, teachers can easily overlook students who need help. Recognizing gazes of students and teachers can be useful in identifying where the students and teachers have their attention in the classroom. To better assess how teachers attempt to guide their students' attention, knowing where the teachers are paying attention by tracking where and what they are looking at during the class time can be useful information. Smart classroom observation systems can be built to record classroom sessions and track gazes of both students and teachers to identify disengaged or distressed students and help teachers to better recognize whether they are paying attention to the right thing or the right student in the classroom.

In this work, we explore a machine learning-based approach to automatic recognition of where a person in the image is looking. In particular, we build an end-to-end deep neural network that takes 2-D static images of multiple people as inputs and infers (x, y) coordinates of where *each* person is looking at in the image as outputs. This computational problem is known as *gaze following* [20]. Gaze following from 2D images is particularly challenging since 1) no additional information of the scene, such as depth information, is available and a person can be looking at any of the different planes of depth in the image, 2) people in the image can be looking at objects either inside the image or outside the image, 3) the eyes of some people may be blurred or partially invisible. Nonetheless, requiring only 2-D images is attractive because of the ubiquity and greater convenience of using commodity 2-D cameras. Our automated system is based on the architecture by [56], who tackled a similar problem for general images from the web. However, our approach differs from theirs in several ways, including the prediction outputs, training algorithm, dataset collection, and application focus.

Figure 1.1 depicts a scenario captured in a classroom observation video. Here, the green arrows are human labeled gaze annotations. In this scenario, a teacher (person #5) is looking at child #3 but not at the disengaged student (child #1), child #1 is looking at something outside the frame and not paying attention to the teacher but other students (child #2, #3 and #4) are paying attention to the teacher's hand. One of the main purposes of this work is to be able to build a computer vision system which can identify the gazes of students and teachers from such classroom observation videos. Gazes of each individual as well as identification of who is looking at whom can be a useful information for identifying attentions of students and teachers which can be an instrumental feature for automated classroom observation systems.

Limitations: Since we are using 2-D static images, one of the biggest limitation is the unavailability of depth information from the image. Therefore, gaze being annotated or predicted by the model can actually be translated to different depths in the real world.

1.1 Contributions of Thesis

There are three main contributions of this work.

1. We build a deep learning-based architecture, based on related work by [56], for automatic eye-gaze following from 2-D images of classroom observation videos. Not only do we focus on each individual's gaze location, we also focus on whom each individual is looking at.

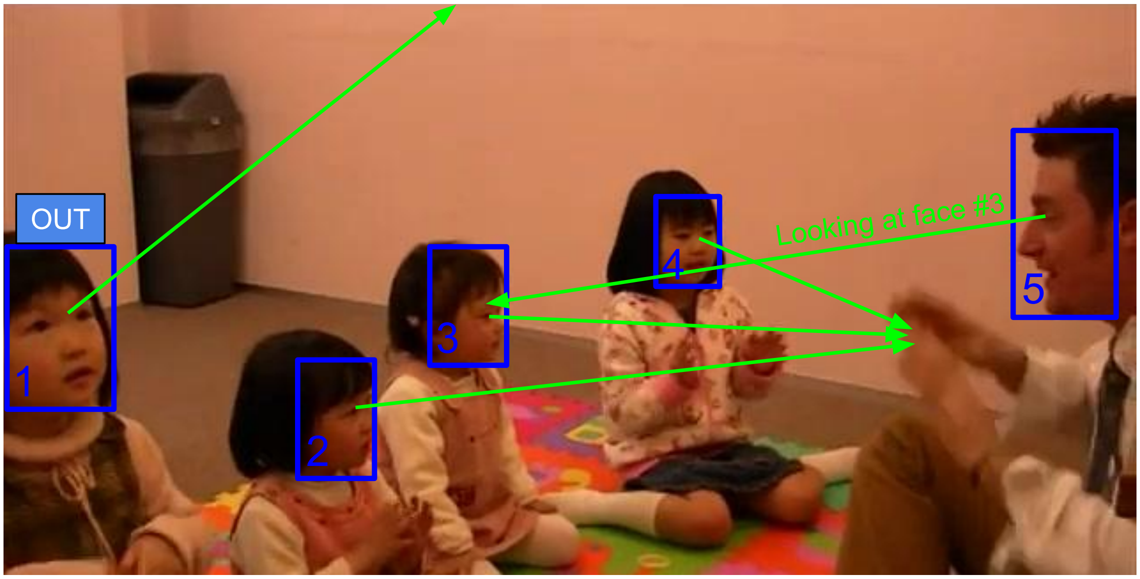


Figure 1.1: Eye gaze targets labeled by a human labeler for each person in the image. Labelers also indicate targets that are located outside the field-of-view (indicated by "OUT"). Can we build a computer vision system that can estimate *where* each person is looking? In this image, the man is looking at child #3. Can we identify automatically *who* each person is looking at? Image from `https://goo.gl/xUdYbC`.

2. We extend the model of [56] to support gaze targets that can be *outside* the camera's field-of-view. Especially due to the lack of depth information, this is a highly challenging problem, both for human labelers and the machine.

3. Since our application focus is on students and teachers in classrooms, we collect and annotate a new gaze dataset which is tailored to our application focus.

1.2 Outline of Thesis

The outlines of the thesis is as follows: in Chapter 2 we review the background work on classroom observations and student-teacher interactions in classrooms. On a more technical side, the background work on deep learning and convolutional neural networks is reviewed since we use CNNs as our building block to build the end-to-end deep learning system for eye-gaze following in classrooms. Then, we present related work on various approaches of following eye gazes in images and video as well as saliency modeling. In Chapter 3, we present data collection process to obtain the eye gaze datasets to train our deep neural networks. In Chapter 4, we present the deep learning model which we built to perform the task of predicting eye gazes of individuals in 2D static images. In Chapter 5 we present how the same network is used to predict which faces the individuals of interest are looking at. We show quantitative and qualitative results of our deep neural network in chapter 6.

Finally, in chapter 7, we conclude and propose future directions which can be taken from this thesis work. The work done in this thesis is intended to serve as a building block for automated classroom assessment systems which can give valuable feedback to teachers.

Chapter 2

Background and Related Work

To understand the approach we took to solve the problem of following gazes of students and teachers from classroom observation videos, it is helpful to know what gaze following is and a brief background on classroom observation videos. We also present background on teacher-student interactions in classrooms and prior work on how states of students, teachers and classrooms are modeled. Knowing the important elements of effective teacher-student interactions can provide crucial insights when building effective computer assisted systems which can provide valuable feedback to teachers [36, 33]. On a more computational side, a quick review of Convolutional Neural Networks (CNNs) and main concepts behind CNNs are presented as CNNs are the main components in building our end-to-end deep learning system for gaze following. Then we present computational approaches previously used to tackle the problem of gaze following using computer vision and more recently, using deep learning. Finally, we present previous work on saliency modeling which has close relations to the concept of gaze following.

2.1 Classroom Observation

A classroom observation is a form of observation by a trained observer to rate the quality of teaching while it is taking place in a classroom or some form of learning environment. As the demand of higher quality of teaching in educational institutions increases, classroom observations have become a common practice to ensure a teachers effectiveness and proficiency in delivering their lessons [6, 47, 60]. Classroom observations are becoming more crucial in teacher performance evaluation, teacher professional development programs and education research as a way to capture not only the quality of the program but also the quality of teacher and his/her methods of teaching [31]. Based on evaluations from classroom observations, schools may use standardize protocols to rate performance of teachers. Researchers also use information from classroom observations to assess features of classrooms that are related to development of student learning and student outcomes [46].

2.1.1 Live Observation and Classroom Observation Videos

Two common ways of conducting classroom observation are: 1) A trained observer performing live observation in the classroom; 2) A video recorded in the classroom during the class session [14]. Live observation has the main advantage of the observer being able to take into account of everything that is happening in the classroom, and not just limited to what is captured within the video frame (e.g. video frame might be fixated to a certain point in the classroom and might miss a crying child who is off the frame). Recorded classroom observation videos are obtained by setting up a video camera to capture the classroom. The video can be recorded either by a teacher him/herself or by a person who records the dynamics of the whole classroom. Recorded classroom observation videos can be obtained more cheaply and does not require a live observer in the classroom. Not only that, classroom observation videos can be assessed by multiple observers asynchronously and therefore, may potentially increase inter-coder reliability.

One of the major difficulties to effectively identify the climate of a classroom and interactions among students and teachers from classroom observation videos is the difficulty to manually code them. Classroom observations usually contain multiple students and teachers interacting simultaneously in different parts of the classroom. Human coders may find it hard to identify every teacher-student interaction, attentions of teachers and engagement of each student. A single human coder can miss subtle but important events in the classroom. Having multiple coders increases cost and evaluations can vary among coders. Therefore, having a computer-assisted classroom observation coding system would partially alleviate some of the problems encountered in coding classroom observations.

2.2 Teacher-student interactions in classrooms

Classrooms are dynamic social settings, whether they contain a single teacher and a few students or multiple teachers with many students, and each classroom can have its own unique teacher-student interactions. The nature and quality of teacher-student interactions have great impact on students' attention, engagement and is predictive of students' development and outcome. Knowing crucial elements in teacher-student interactions via standard observation methods can provide teachers with personalized feedback about their interactions. Such feedback is shown to promote better teacher-student interactions and increases student's engagement [52]. Studies have also shown that there is a relationship between emotional and instructional support provided by teachers and student's cognitive, social and emotional skills [36, 46]. Researchers often conduct classroom observation sessions to discover how teacher-student interactions impact student's outcomes. One particular example is the Gates Foundation Measures of Effective Teacher (MET) project [33] where tens of thousands of hours of classroom observation videos are recorded with the aim of discovering most effective methods to teach students.

One important element of effective teacher-student interaction, which is related to this work, is the eye-gaze of teachers and students. Knowing where teachers are looking at in the classroom is a good indicative feature to identify where they are paying attention. Identifying whether teachers pay attention to disengaged students, students in need of help or students who are being bullied by other students (teacher sensitivity) are important factors to consider when providing crucial feedback to teachers. Knowing eye gazes of students can provide information of students' attention and intentions [11].

2.3 Modeling states of students, teachers and classrooms

There has been much prior work done in modeling states of students engaged in a learning task using various Intelligent Tutoring Systems. Various computer vision techniques are employed to analyze facial features to identify affective states and engagement levels of students [26, 10]. Some studies [34] examine whether making use of multi-modal data (computer vision for facial expression recognition and array of other sensors) can be beneficial in picking up non-verbal cues which are indicative of learner's frustration. Researchers also started to use muti-modal data combined with machine learning to capture the dynamics of the entire classroom. Speech recognition technologies have been widely used to capture various aspect of classrooms, such as classifying classroom activities [67] and modeling teachers during their lectures [19]. Both speech recognition and computer vision are also deployed for automated analysis of teacher-student interactions in live classrooms. [18].

In this work, we build a computer-vision system using deep neural networks for *automated eye gaze following* that estimates, for each person in the classroom, where she/he is looking. Such a system can be used to data-mine classroom observation video datasets. It could also facilitate "smart classrooms", which track gazes of both students and teachers, identify disengaged or distressed students, and help teachers to better recognize whether they are paying attention to the right thing or the right student in the classroom.

2.4 Deep Learning

Machine learning algorithms are data-driven in nature and the success of machine learning methods depends on learning good representations of data which is used to train machine learning algorithms. Deep learning, a subfield of machine learning, is primarily focused on learning good representations of training data by hierarchical structures (Figure 2.1). Given hundreds of thousands of images as training data, multiple layers of non-linear functions transform the input data into lower-level, concrete representations (e.g. edges, corners, colors, etc.) in earlier transformations and into higher-level, more abstract concepts (e.g. shape of a face, shape of a wheel, etc.) in later transformations. The parameter of transformations are learned rather than designed. This contrasts with hand-engineered filters (e.g. Sobel filter for edge detection) or hand-engineered feature descriptors (e.g. SIFT [44], HOG [15]) used in traditional approaches in the domain of computer vision. In deep learning, the error between the true label and the predicted output from the algorithm is back propagated through the deep network and each parameter is corrected by a small step which would make the error smaller. This approach gave birth to one of the most effective machine learning architectures called artificial neural networks [58]. High-level pipelines for object recognition in images using traditional computer vision approaches and using deep neural networks are shown in Figure 2.2.

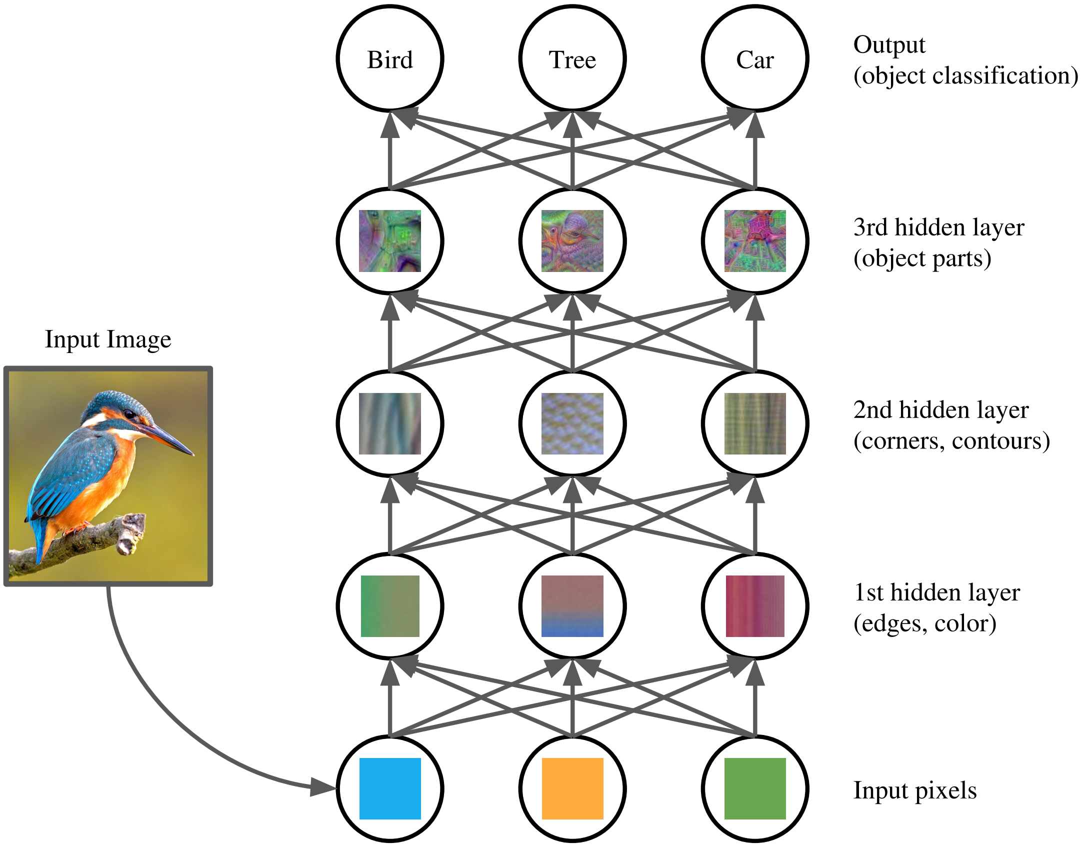


Figure 2.1: Illustration of how deep neural networks learn hierarchical structures. Image reproduced from [25]. Kingfisher image taken from [68]. Filters taken from [64].

Each layer of artificial neural networks contains multiple *neurons* or *nodes* that calculate the weighted linear combination of the outputs of the previous layer, which are then fed into a non-linear activation functions such as Rectified Linear Unit (ReLU) $f(x) = max(0, x)$ [49] or Sigmoid $f(x) = \frac{1}{1+e^{-x}}$ to produce a single scalar output. The parameters of the network are the weights between each neurons and the neurons in the layer below. Neural networks with multiple layers are called deep neural networks and deep learning is a class of machine learning algorithms that use deep neural networks to learn the representations of training data to make predictions on the unseen data. Deep neural networks have become more powerful due to the availability of more computational power, availability of more training data, and advent of novel deep neural network architectures.

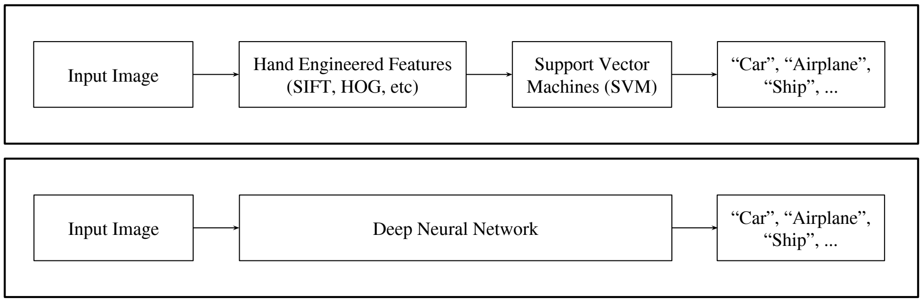

Input Image
Hand Engineered Features (SIFT, HOG, etc)
Support Vector Machines (SVM)
"Car", "Airplane", "Ship", ...
Input Image
Deep Neural Network
"Car", "Airplane", "Ship", ...

Figure 2.2: **Top**: Traditional computer vision approach used for classifying objects. Interest points are detected with hand-engineered feature descriptors (e.g. SIFT, HOG) and classifiers such as Support Vector Machines (SVM) are used to learn and classify objects based on extracted features. **Bottom**: Deep neural network approach used for classifying objects. Deep neural networks learn representations of objects in images from multitude of training data in a completely end-to-end fashion. This eliminates the need of hand-engineered features.

2.4.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) [41] are a class of feed-forward deep neural networks which are primarily used for computer vision related tasks such as object classification, object recognition and image segmentation. In fully connected feed-forward neural networks, the input data is vectorized before fed into the network and every node in every layer is connected to every other node in preceding layer. One of the biggest drawbacks of regular feed-forward neural networks is that it does not scale well to images of various sizes and the spatial structures of images are not preserved. With CNNs, the network is forced to use the spatial structure of an image and the fundamental principle behind CNNs take advantage of the spatial structure in images by reducing the number of free parameters in the network as much as possible without overly reducing its computational power. By reducing the number of free parameters, probability of correct generalization increases because having less free parameters not only reduces the entropy of the network [12, 17, 65] but also reduces the Vapnik-Chervonenkis (VC) dimensionality [4].

Due to the fast pace of research and development of CNN architectures, many different kinds of CNN architectures have emerged but a typical CNN consists of a convolution layer, pooling layer and fully connected layer as fundamental building blocks. One of the earliest and well-known handwritten character recognition CNN 'LeNet-5' [40] is depicted in Figure 2.3.

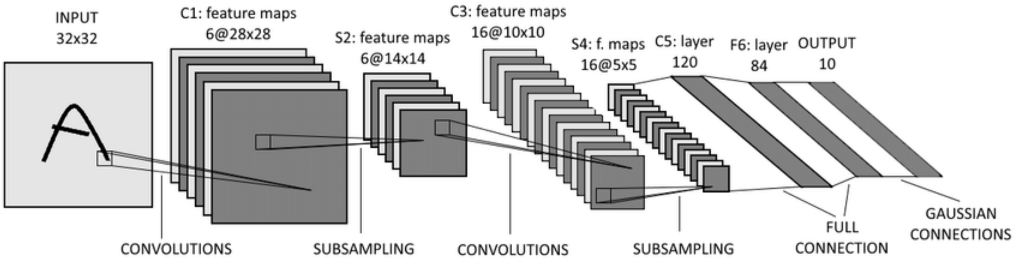


Figure 2.3: Convolutional Neural Network 'LeNet-5' used to recognize handwritten digits. Image taken from [40].

Convolution layers: As the name suggests, convolution layers perform linear operation called *convolution*. The operation of *convolution* in CNNs has differences with the definition of convolution in other fields. Convolution makes it possible to build CNNs which can accept images of different sizes. The input of a CNN is usually a 3-D tensor of an RGB image ($width \times height \times colorchannel$) or just 2-D tensor of a greyscale image ($width \times height$) and the kernel is usually a 2-D tensor of learnable weights. Usually, kernels of convolution layers are much smaller than the dimension of the original image. The convolution operation can be mathematically defined as:

$$s[i,j] = (I \times K)[i,j] = \sum_m \sum_n I[m,n]K[i-m, j-n]$$

where I is the input image with dimension $m \times n$, K is the kernel and s is the output at location i and j. Each convolution layer of a CNN usually have multiple kernels. Output of convolution operation is fed into a non-linearity function (e.g. ReLU) before it is fed into a pooling layer.

One important aspect of CNNs is that the convolution layers share parameters. In a regular feed-forward neural network, each elements of the weight matrix is used exactly once when output layer is being computed. When convolution is applied to images, convolution create 2-D feature maps that show where certain features appear in the input image. For example, if the feature map filters vertical edges in images, the same filter can be applied to the whole image since the image may contain multiple vertical edges all over the image. This form of sharing the same kernel for the whole input data is known as parameter sharing (or weight sharing) and this makes convolution layers *translation equivariant* for input data.

Pooling layers: Pooling layers reduce the dimension of input data by replacing the neighborhoods of output units with average or max value of all the neighborhood cell values. The operation of taking the average value when pooling is known as 'average pooling' and the operation of taking the max value when pooling is known as 'max pooling'. One important detail of the pooling layer is the stride of the sliding window and parameter specified for stride determines whether the sliding windows overlap with one another or not. For example, if the input of a pooling layer is the size of 16×16 and the pooling window is the size of 2×2 with stride of 2, the pooling window slides the across the input in the fashion of a sliding window and summarizes every 4 cells covered by the window by taking the average or max values. Since the stride is 2, the sliding window moves to next region without having any overlaps with the previous region. In this fashion, either max pooling or average pooling will produce an output of 8×8. However, [38] shows that overlapping regions reduce overfitting. For the example described above, pooling with overlaps can be achieved by setting the stride to 1. By doing so, pooling operation will reduce the original input of 16×16 to output of 15×15.

Pooling helps to make the representations *invariant to small translations* [25]. Invariance to translation means that the values of most of the pooled outputs would not change just by translating a small amount in the input. Invariance to translation is a useful property if the location of certain features should appear exactly at particular locations in the object.

Even though the concept behind pooling layers is to lower the dimension of input data, new concepts such as global average pooling [42] and global max pooling are used for different purposes. Global average pooling takes the average of each feature map, and the resulting vector is fed directly into the softmax layer. Global average pooling essentially replaces the traditional fully connected layers after convolution layers in CNNs and acts a structural regularizer which prevents overfitting globally [42].

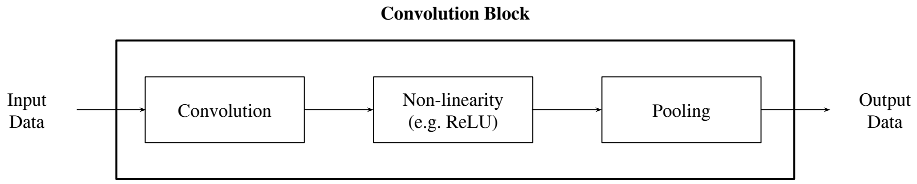


Figure 2.4: Components of a typical convolution block.

2.4.2 Deep CNN Architectures

In this section, background of some of the most influential deep CNN architectures are presented. Deeper and more sophisticated deep CNN architectures are being developed at a rapid rate by building upon more sophisticated concepts and harnessing the availability of more powerful hardware tailored to deep learning.

AlexNet: AlexNet [38] is one of the first architectures which popularized CNNs in computer vision. AlexNet trained on ImageNet dataset achieved a winning top-5 test error rate of 15.3% in ILSVRC-2012 competition. Compared to LeNet-5 [40], AlexNet is a deeper architecture containing 5 convolutional layers, 3 max pooling and 3 fully-connected layers and has 60 million parameters. The architecture of AlexNet is depicted in Figure 2.5. AlexNet popularized several elements found in typical CNNs such as ReLU, overlapping pooling, dropout and augmenting training data. However, normalization method used in original AlexNet, Local Response Normalization, is less common in recent CNNs.

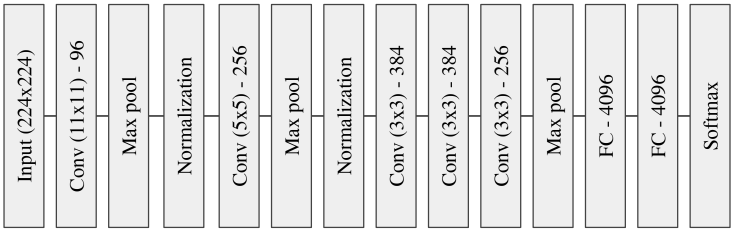


Figure 2.5: AlexNet architecture.

VGGNet: VGGNet architecture (2014) [61] uses only 3×3 convolution filters for all the convolution layers. The reason behind using smaller convolution filters is that stacking smaller convolution layers can emulate the effect of larger convolution layers while increasing the depth of the network. For example, without having any spatial pooling layers in between each convolution layers, stacking two 3×3 convolution layers effectively emulates a 5×5 convolution layer and 7×7 convolution layer can be obtained by stacking three 3×3 convolution layers.

VGGNet has two variants: VGG16 and VGG19 (consisting of 16 and 19 convolution layers respectively). VGGNet is appealing due to its uniform architecture while being able to achieve state-of-the-art results in image recognition tasks. In VGGNet, spatial pooling is carried out by five max pooling layers. The difference between typical convolution blocks (as depicted in Figure 2.4) and convolution blocks in VGGNet is that pooling layers do not immediately follow every convolution layers. At every pooling layer, max pooling is performed over a 2×2 window, with stride 2. One main drawback of VGGNet is that there are 138 and 144 million parameters for

VGG16 and VGG19 respectively. An architecture of this size is prone to overfitting if the training data size is small. Depiction of a VGG16 is shown in Figure 2.6.

In this work, VGG16 is used as the base CNN architecture but combined with concepts from other architectures such as Global Average Pooling [42] to dramatically reduce fully connected layers and thus reducing the risk of overfitting and time required to train.

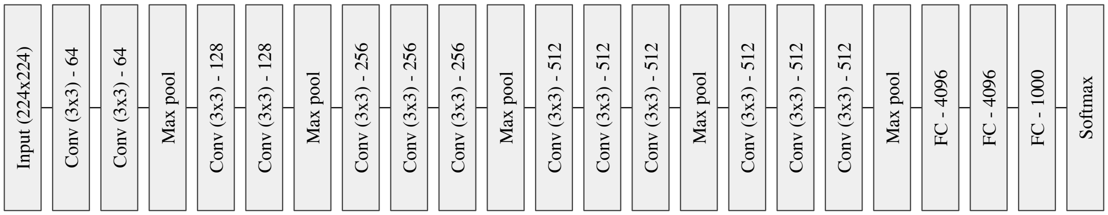


Figure 2.6: Depiction of VGG16 architecture. n in each Conv (3×3) - n refers to the number of filters in that convolution layer. FC-n refers to fully connected layer with n nodes in that layer.

Inception: The most straightforward way to improve the performance of deep neural networks is to increase their size, both depth (number of hidden layers) and width (the number of neurons in each layer). This approach works well when there are tens of millions training data but downside of having a deep neural network with more parameters is that it is more likely to overfit when the training data size is small. Not only that, deeper and wider networks take more computational resources to train as well. With the intention of being able to build deeper networks without blowing up the number of parameters or incurring heavy computational cost, Inception [63] architecture introduces building blocks called 'Inception Modules' (Figure 2.7). Inception architecture consists of inception modules which are stacked upon each other with occasional max pooling layer in between the inception modules to reduce the resolution of input data. Inception architecture introduces inception modules only in deeper levels while keeping traditional convolutional layers in earlier levels. In order to keep computational requirements manageable, 1×1 convolution layers are introduced in inception modules. Their purpose is to reduce input dimension by reducing the number of filters from incoming data (e.g. reducing 128 filters to 64 filters before being fed to 3×3 and 5×5 convolution layers in Figure 2.7). GoogLeNet which is based on Inception architecture contains 22 weight layers but has 12 times less number of parameters compared to AlexNet [38].

2.4.3 Transfer Learning

CNN architectures such as AlexNet [38], VGGNet [61], Inception [63] and ResNet [27] are trained on large datasets such as ImageNet [59] which contains over 15

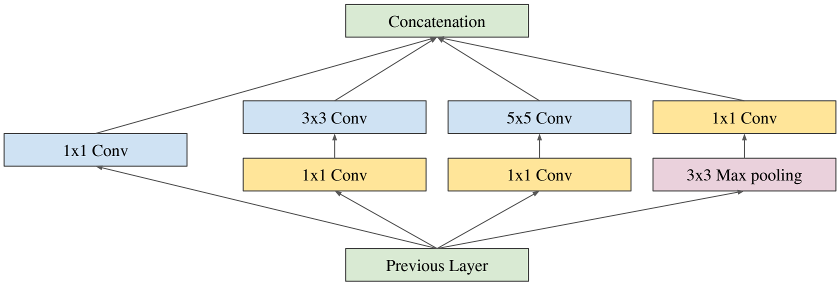


Figure 2.7: Inception module with dimension reductions. 1×1 convolutions reduce the computational bottleneck by reducing the dimension of input data. This allows deeper and wider network architecture without having much impact on computational cost. Image reproduced from [63].

million labeled high-resolution images belonging to roughly 22,000 categories. CNN architectures which contains tens or even hundreds of millions of parameters to be tuned are difficult to train since those networks can overfit easily if the training dataset is not large enough.

[50] showed that internal convolution layers of CNNs, which are pre-trained on large datasets such as ImageNet for object classification tasks, can extract mid-level image representations and these image representations can be used in different image classification tasks. Convolution layers might have learned kernels for edge detection, corner detection and color blob detection during their learning phase. These kernels are useful for other object classification or other computer vision related tasks and reusing these kernels is more effective and efficient rather than relearning from scratch. Therefore, even in the case when the image statistics of original large image datasets and smaller image dataset for actual training are different, the representation that convolution layers learnt from the larger dataset leads to significantly improved results for object classification and recognition when transferred to the convolution layers of a CNN which will be trained on smaller training dataset. The technique of transferring the weights of convolution layers which are trained on a different dataset to a new dataset is called *Transfer Learning* and it is an instrumental technique to train large CNN architectures on relatively small datasets.

Fine-tuning: A strategy when training a CNN with transfer learning is fine-tuning the convolution layers to accommodate statistics of new datasets. The process of performing transfer learning involves: 1) attaching new fully connected layers and 2) fine-tuning convolution layers.

Fully connected layers from the original CNN are chopped off since weights learnt in fully connected layers are task specific and those weights are only useful for the original task but may be irrelevant for the new task. After chopping off the fully connected layers, new fully connected layers with random weights are attached. The initial phase of training involves freezing the convolution layers and only training the newly attached fully connected layers until the output loss is saturated or the network starts to overfit. Then the convolution layers are unfrozen and the whole network is trained with lower learning rate.

In this work, we use transfer learning by transferring the weights of convolution layers of VGG16 pre-trained on ImageNet. Fully connected layers and softmax layer from the original VGG16 are chopped off (four rightmost layers in Figure 2.6) and new fully connected layers initialized with random weights are attached. During the initial training phase, only fully connected layers are trained while keeping the convolution layers frozen. After the loss is saturated, convolution layers are unfrozen and the whole network is trained with lower learning rate.

2.5 Eye gaze following

Due to the richness of information that the human face can provide, researches have tried to understand the perceptual, cognitive and neurological processes which can be extracted from human face [24]. One particular important aspect of human faces is the eye and the direction of eye gaze. The direction of gaze is a good indicative feature of what the attention of a person is [3, 54] and mutual gaze (eye contact) provides the main mode of establishing communication between humans [35, 62]. Direction of gaze plays a crucial role in learning and development in children [2, 29], in daily social interactions [20, 7] and in joint attention among people [66, 7]. [20] coined the term "gaze following" as the act of a person following the line of sight of another person. Gaze following is considered to be sociocognitive behavior which serves numerous functions such as learning, collaboration, coordination, understanding others' intentions and directing others' attentions [20].

2.5.1 Eye gaze following with computer vision and deep learning

Due to the importance of following gaze of others, which humans do naturally when communicating, collaborating and socializing, researchers in the field of robotics, computer vision and machine learning have recently started to formulate and tackle the problem of automatic gaze following within different contexts. Effective estimation of gaze direction involves two main components: (1) head pose and (2) direction of eye gaze [51]. Therefore, a computer vision model which does full gaze direction estimation should address both components. Gaze-following is used in [51] to improve models of free-viewing saliency prediction. However, they only estimate the gaze direction without identifying the object being attended.

In some settings [32, 9, 22], there is only a single person whose gaze is being followed, e.g., a student who is interacting with a mobile phone or tablet [37] to play an educational game [71]. [22] presents a method for detecting and recognizing social interactions from ego-centric cameras by computing lines of sight into locations in space to which individuals attend. [23] used videos recorded from ego-centric cameras to recognize daily activities by combining gaze fixation, visual features and action labels using probabilistic generative model. Deep learning has been applied to eye-tracking specifically for mobile devices as well.

In other settings (such as ours), the camera examines an entire scene containing many people, and the gaze of *each* person in the scene is followed [48] [45] [57] [56]. While most of the prior work uses RGB data, [48] approached the problem of modeling human attention by using convolutional neural networks to estimate head pose and gaze direction from high resolution RGB-D data. Their approach uses multi-modal RGB-D data to first classify gaze directions of individuals over 8 classes spanning 360 degrees around their head and fine-tune a regressor based on the learned deep classifier to provide finer gaze directions. If multi-modal RGB-D data is unavailable (e.g. low-resolution surveillance video), their method can fall back on classifying the head pose into 8 equally spaced bins around individuals' heads. [45] used head pose estimates to identify interactions between people in movies by detecting whether they are looking at each other or not. But the scope of their work is limited to people looking at each other in the same scene in contrast to people looking at other people in different scenes or people looking at objects. More recently, researchers have considered gaze following not only from static images, but also how to harness temporal information from an entire video to better estimate the person's gaze target [57].

Our work is based on the work by Recasens, et al [56]. In their work, they created GazeFollow dataset which contains gaze annotations of people doing various actions in different environments. Using this dataset, they trained convolutional neural networks to track both gaze direction and gaze location of each person in the image. But they limit the scope of gaze following by focusing only on people who are looking at objects or people inside the images.

2.6 Saliency modeling

Gaze following is related to saliency modeling, whereby image features of different levels of abstraction (low-, mid-, and high-level) are examined to consider the most likely locations in the image to which an observer would visually attend [32]. [9] made a connection between these two by stating that an observer looking at an image containing people may follow the gaze of people rather than actually fixating on salient objects in that image. Therefore, gaze following can play a complementary role in solving the problem of saliency model of attention. [16] explored the problem of predicting a driver's gaze behaviours and identifying the attention of a driver by detecting saliency in a complex driving environments. A comprehensive survey by [8] reviews key concepts, core techniques, datasets and evaluation metrics used in salient object detection.

Chapter 3

Data collection

In order to train and evaluate our end-to-end gaze following deep learning system, we need to have a sufficient amount of annotated data for the task. Since the application focus of our work is gaze following in *school classrooms*, we collected our own dataset of classroom observation sessions. In particular, we used 70 videos publicly available on YouTube of school classrooms [55]. In contrast to publicly available annotated data on gaze following (the only such dataset of which we are aware is GazeFollow [56]), classroom observation videos often contain *many* people per image frame, and the kinds of background clutter differ significantly from that of GazeFollow, which largely consists of images used for more general object detection research such as: SUN [69], MS COCO [43], Actions 40 [70], PASCAL[21], ImageNet detection challenge [59] and Places dataset [73]. Since we would like to focus on identifying the gaze of students and teachers in classrooms, we decided to collect our own dataset consisting of scenes from school classrooms. The process of collecting the required datasets for training our neural networks is detailed in the following section.

3.1 Data Collection Process

Steps for data collection process are shown in Figure 3.1. We begin with obtaining 70 classroom observation videos publicly available on YouTube curated by [55]. The classroom observation videos contain pre-school students engaged in interactive learning settings or traditional classrooms settings. Static frames are extracted from those videos at a rate of 1 frame approximately every 10 seconds. After extracting frames from videos we used faster R-CNN for face detection [30] to obtain face bounding boxes (top left (x, y) coordinate, width and height) in extracted frames. Extracted frames contain 8 faces per image on average. Face bounding boxes are overlaid on images during the annotation step. During the annotation step (Section 3.2), multiple labelers are asked to draw gazes of individuals identified by face bounding boxes obtained in previous step. After all the data has been annotated, the quality of annotations are assessed by performing inter-labeler reliability tests

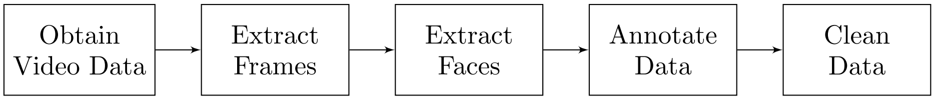


Figure 3.1: Data collection process

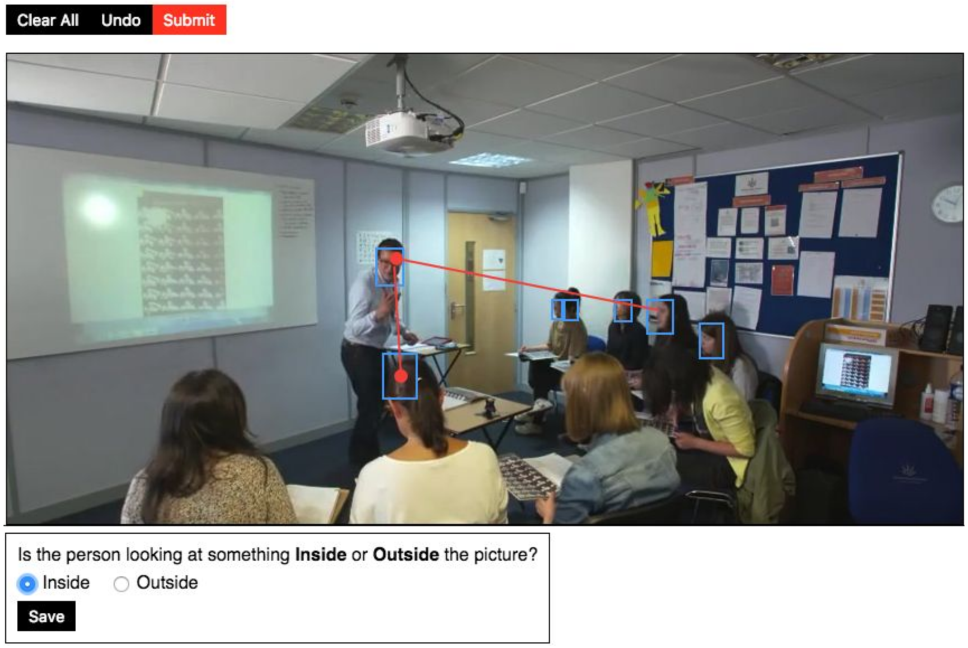
(Section 3.3) and visualizing multiple annotations on original images. Low quality gaze annotations are culled out from the original annotations. Finally, gaze dataset which is ready to be consumed by the deep neural networks is created. 70% of the dataset is used as training set, 15% as validation set and the remaining 15% as test set. Each data split contains different classroom environments as well as different individuals.

3.2 Details of Annotation Process

Ground-truth gaze annotations from the image frames were collected using at least 3 labelers per image on Amazon Mechanical Turk (AMT). Labelers used an online annotation tool that we custom-built from scratch using JavaScript and HTML5 for this work, to annotate two main components of each subject in each scene. The first component is to identify the gaze target for each person (identified automatically by the face detector as described in Section 3.1). The gaze is indicated by a line, starting between the eyes of a person and ending on an object or a person which the person is attending to (the end of gaze is depicted by filled red circle - Figure 3.2). The second component is the indication of whether the person is looking at something inside or outside the image. Labelers indicate whether the gazes end inside the image or extend outside the image by choosing the appropriate option in the annotation tool depicted by a radio button below the image in Figure 3.2. We collected three gaze annotations, each for $17,758$ faces in $2,263$ images, resulting in a total of $48,907$ gaze annotations from 408 unique annotators. Some of the faces are labeled by more than three labelers. Samples of gaze annotations collected from AMT can be seen in Figure 3.3.

3.3 Inter-labeler Reliability Tests

In order to assess how well the machine performs compared to humans, we thus need to measure human annotators' accuracy in a comparable task. Since we ask multiple labelers to annotate gaze locations of each face in each image, we have to assess how well humans agree with each other in labelling the gazes from 2D static images. Due to the relative difficulty of the annotation task, we observe that humans have different opinions on annotating where a particular face in an image is looking at (e.g. Figure 3.4 (b)). Even though there are many approaches which can be taken to assess inter-labeler reliability, we present two approaches which we



Is the person looking at something **Inside** or **Outside** the picture?

● Inside ○ Outside

Figure 3.2: Custom-built annotation tool deployed on Amazon Mechanical Turk. Labelers use this tool to draw an eye gaze of a person from the eyes of that person to an object or a person of attention. A labeler draws the gaze of a subject in the image starting within the bounding box of that subject and ending where the gaze of that subject ends (depicted by a filled circle). If that subject is looking at something outside the image, the labeler has to choose "Outside" in the selection box to indicate that the gaze ends outside the image. Image from `https://goo.gl/gWu4P5`

use for this work in the following two subsections.

3.3.1 Inter-labeler reliability test on 256×256 pixel images

We take the original (x, y) coordinates of annotated gaze end points to observe how well humans are doing in terms of mean Euclidean distance between each other. Since we each image in the dataset with three labelers, we can treat each human labeler in turn as a prediction while treating the mean of the remaining 2 as a ground truth. By doing so, we can quantify how well one labeler performs with respect to the remaining 2 labelers. A single labeler is able to achieve mean Euclidean distance of 41.04 on 256×256 pixel input images. Using the same procedure for calculating the inter-labeler mean Euclidean distance, we additionally calculated mean absolute error (MAE), mean absolute angular error and area under the receiver operating characteristic curve for indication of whether the gazes end inside the image or end outside the image (AUC for In/Out) (shown in Table 3.1). Mean euclidean distance serves as a metric for determining how well the labelers agree on where the gaze should end in the image and mean absolute angular error serves as a metric for

Table 3.1: Inter-labeler reliability scores calculated on 256×256 pixel images. Gaze end points are treated as (x, y) coordinates.

MAE (pixels)	Mean Euclidean Distance (pixels)	Mean Absolute Angular Error	AUC for In/Out
25.91	41.04	18.38	0.70

determining how well the labelers agree on the direction of the gaze. Mean Euclidean distance and mean absolute angular error calculated among multiple labelers serve as benchmarks for our automatic eye-gaze following deep neural network.

3.3.2 Inter-labeler reliability test on $N \times N$ grid

As we will describe later, one of the two machine learning-based approaches that we use for automatic eye-gaze following is based on dividing up the 2-D classroom image 8×8 grid cells. Grid size of 8×8 is the largest number of grid cells N such that the size of each cell (measured diagonally from two opposite cell corners) was no smaller than the inter-human reliability measured in pixels (41.04 pixels over 256×256 pixel image). We calculated Inter-labeler reliability using Fleiss' Kappa [39]. Fleiss' Kappa is one of the ways to assess the reliability of multiple labelers more than two performing a classification task and is used to calculate the degree of agreement in classification task (< 0.0: Poor agreement, 1.0: Perfect agreement).

For this reason, the (x, y) coordinates of gaze locations are converted to cells on quantized $N \times N$ grid (Figure 3.4 shows an example of 8×8 grid). Two labelers are regarded to be in agreement when (x, y) coordinates of their gaze annotations fall in the same cell of the $N \times N$ grid. By using this convention, we calculated inter-labeler reliability using Fleiss' Kappa. The inter-labeler reliability among three labelers for gaze annotations which are inside the image is $\kappa = 0.2639$ for 8×8 grid (Fair agreement [39]). Table 3.2 compares Fleiss' Kappas with different grid sizes. Smaller number of cells have higher Fleiss' Kappa score since gaze annotations from different labelers are more likely to exist in the same cell. Figure 3.4 shows gaze drawings of different labelers on 8×8 grid.

Table 3.2: Fleiss' Kappa for gaze locations on quantized $N \times N$ grids. Gazes are treated as cells on 8×8 grid. The inter-labeler reliability scores are calculated only for gazes which all labelers unanimously agree that those gazes are inside images (Labels indicated as "Inside" during annotation process).

Num Labelers	Frame Count	N=3	N=4	N=5	N=6	N=7	N=8
1	636	1.0	1.0	1.0	1.0	1.0	1.0
2	2,529	0.44	0.38	0.32	0.29	0.25	0.22
3	9,804	0.46	0.41	0.35	0.32	0.28	**0.26**
4	134	0.38	0.31	0.29	0.24	0.21	0.18
5	5	0.38	0.38	0.41	0.17	0.18	0.27

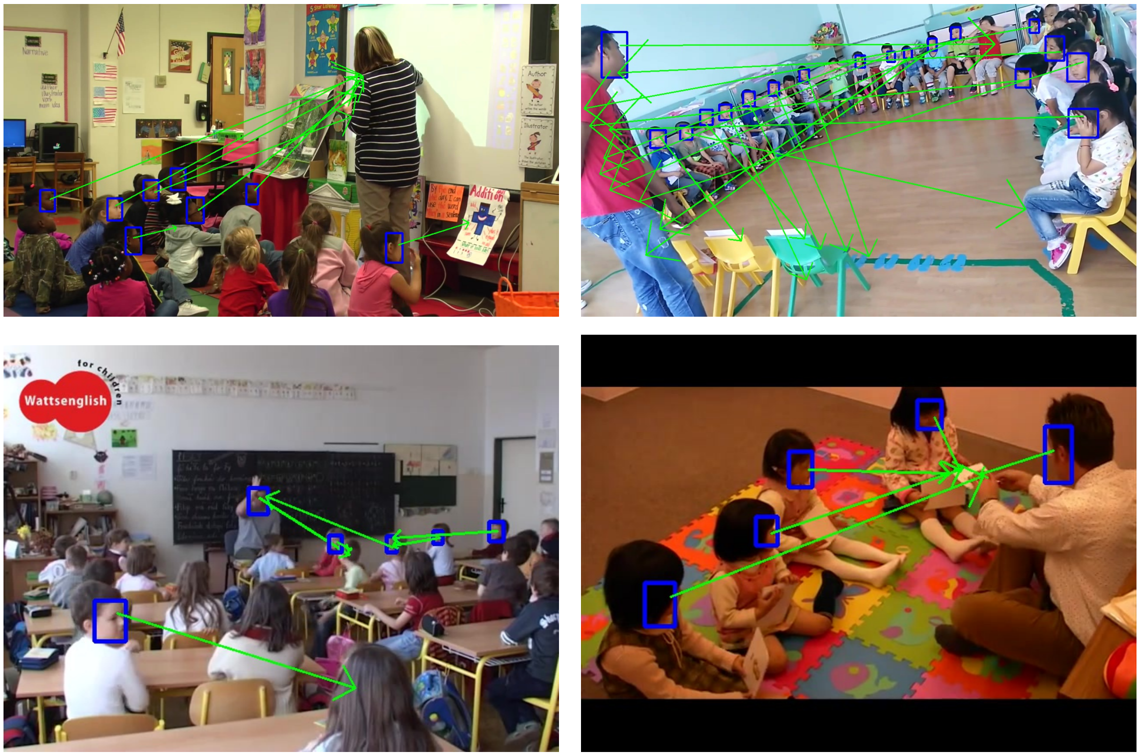
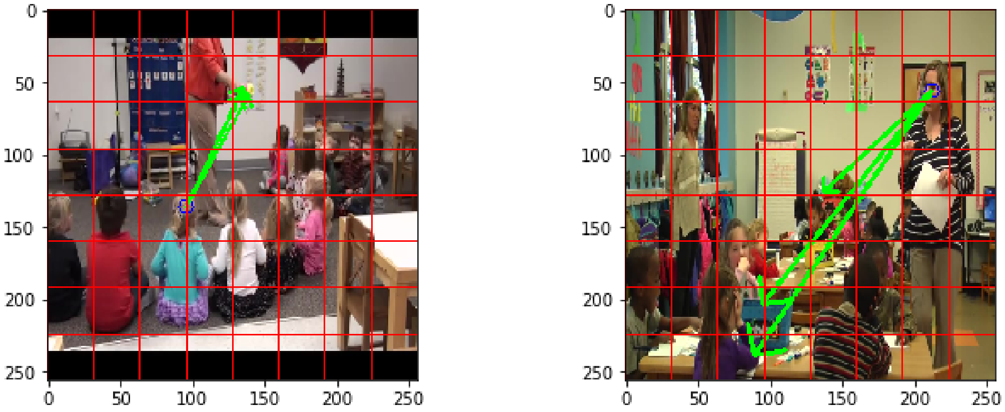


Figure 3.3: Gaze annotations collected using multiple labelers on Amazon Mechanical Turk. Static image frames are extracted from YouTube classroom observation videos [55]. Images (in clockwise direction) taken from: `https://goo.gl/pb19gi`, `https://goo.gl/rCVudX`, `https://goo.gl/xUdYbC`, `https://goo.gl/wbdfZF`.



(a)

(b)

Figure 3.4: Gaze annotations from different labelers for a single face plotted on 8×8 grid. All three labelers highly agree (all gazes end in a single cell) on where the gaze ends for a single face in (a). Each labeler has a different gaze annotation (all gazes end in different cells) for a single face in (b). Images (left to right) taken from: `https://goo.gl/J2jMXo`, `https://goo.gl/rCtU1e`

Chapter 4

Building Gaze Following Network

In this chapter, we present implementation details of the deep neural network based on work done by [56] to predict gaze locations and indication of whether the gazes end inside or extend outside the image. Even though the approach is based on [56], we present differences we made in terms of architecture choice and implementation details to suit our application focus.

4.1 Approach

Using the datasets annotated on AMT, our goal is to build a convolutional neural network (CNN) which takes in the whole image of the scene and predicts the gaze target of each person in the image along with the indication of whether that target is inside or outside the image. We have observed from our annotated datasets that predicting gaze can be ambiguous. If there are multiple people or several salient objects in the image, or the eyes of individuals in the image are not clearly visible, human labelers may disagree when predicting gaze locations. Due to this inherent uncertainty in the problem, we are exploring various options to design our model to support multimodal predications.

One of the primary purposes of this thesis work is to build a computer vision system which can provide gaze locations of individuals in classrooms to an automated classroom observation system which consumes this information for some form of downstream processing. Therefore, we explore both options of treating gaze following as a regression task or as a classification task. If the requesting system requires fine grained information, regression approach would be more useful since our system will be able to provide exact (x, y) coordinates of the gaze. On the other hand, if the requesting system only requires heatmap of a person's attention, classification approach would be more useful.

4.1.1 Regression

When treating gaze following as a regression task, the network regresses to (x, y) coordinates of the gaze target of each person in the image using Euclidean distance between the predicted and ground-truth as the loss function. The disadvantage of using regression is that our predictions are constrained to be unimodal. Since each face in each image was labeled by multiple annotators, we can define the ground-truth by either (a) computing the mean (x, y) location over all labels per face, or (b) treating each location as a separate label. In this work, we treat each gaze location as a separate label when regression approach is taken.

4.1.2 Classification

When treating gaze following as a a classification task, the gaze location is first quantized into a cell on a $N \times N$ grid, and the network's job is to choose the correct cell which contains the gaze end point for each person in the image. As the loss function we can use cross-entropy loss. Classification naturally supports multimodal outputs since multiple gaze annotations at different cells can be treated as soft labels [1]. Figure 4.1 depicts how multiple annotations are converted to soft labels on 8×8 grid. The disadvantage of this approach is that the choice of grid size can affect precision of predictions (i.e. smaller grid sizes will result in poorer precision). In this work, we choose 8×8 as the grid size for classification approach (choice of grid size 8×8 is explained in Section 3.3.2). Exploring different grid sizes to observe the effect of model's performance can be pursued as part of the future work. Another issue with classification approach is that cross-entropy loss does not gradually penalize spatial classes – misclassification which is off by one grid cell is penalized just as much as misclassification which is off by several cells on a grid.

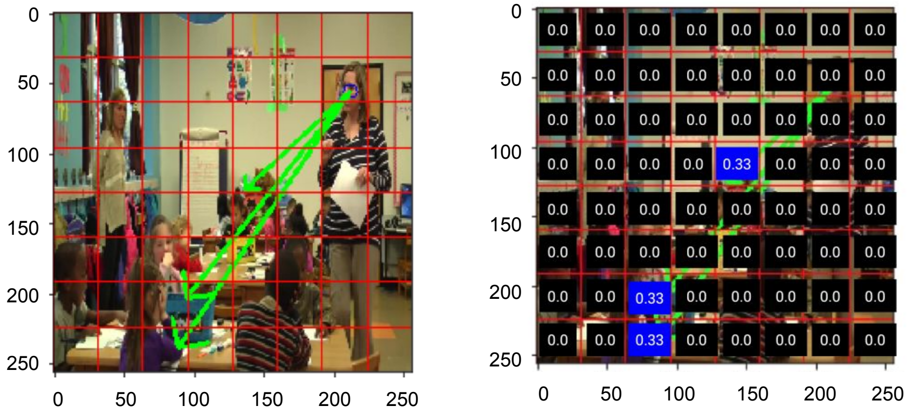


Figure 4.1: Gaze prediction with classification approach. **Left:** Original image resized to 256×256 pixels and overlaid with 8×8 grid. **Right:** Soft labels associated with gaze end points by three labelers. Image taken from `https://goo.gl/rCtU1e`

4.2 Architecture

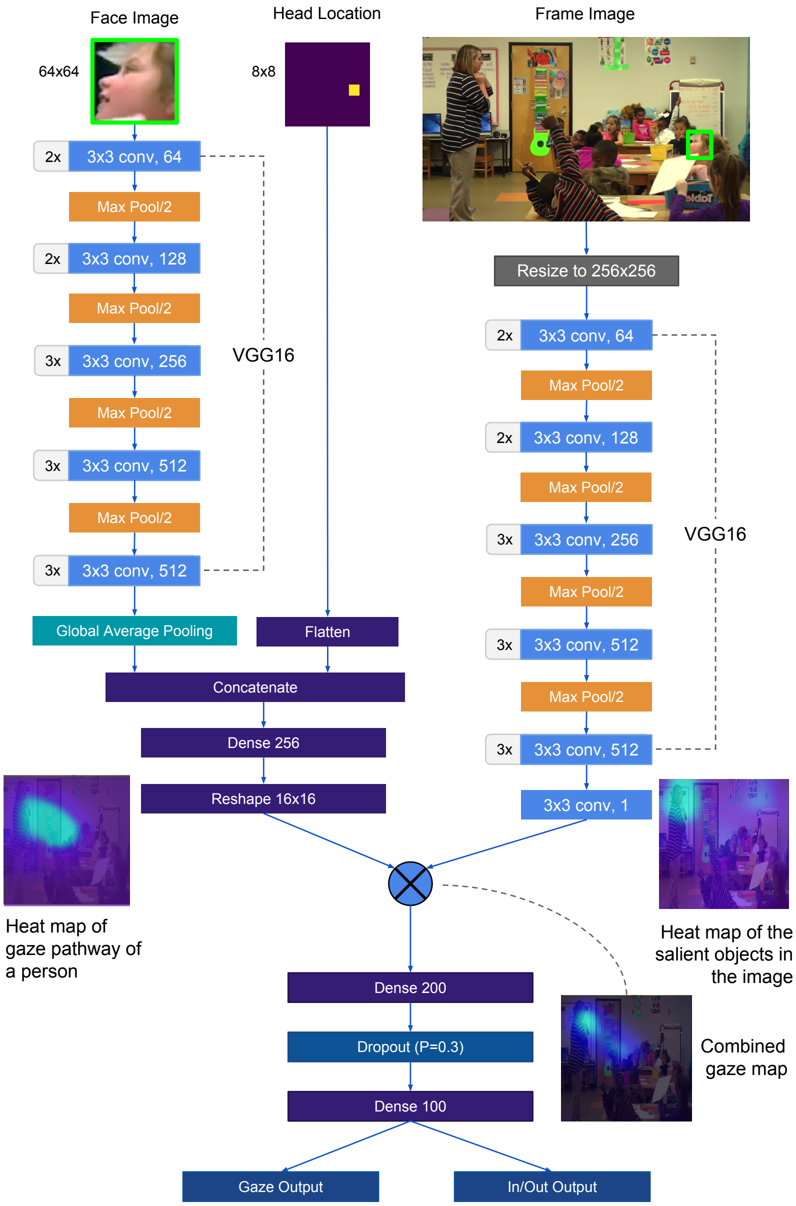


Figure 4.2: Deep neural network architecture, based on [56], for automatic eye-gaze following in school classrooms, consisting of two independent prediction pathways. Each subnetwork produces an independent prediction of the gaze target for each person in the image; the predictions are merged by pixel-wise multiplication to predict both the location of the gaze target and whether the target is inside/outside the image.

The deep learning architecture is based on the model by [56] and is depicted in

Figure 4.2. The gaze target for each person in the image is predicted independently based on two information sources: close-up information of the person's face (automatically detected by a separate face detection network [30]), and the the whole image. Each information source is processed by a separate pathway consisting of a CNN, and the pathways' predictions about the person's gaze target are merged at the end. We call the combined architecture the *Merged Model.* In contrast to [56], we use the VGG16 [61] as the backbone of each CNN because we empirically found that it performed better than AlexNet [38] which is originally used by [56].

Inputs: The inputs of the Merged Model are a cropped, close-up face image (64×64 pixels); the $(r, c) \in 8 \times 8$ location of the center of the person's head in the image for classification approach or (x, y) coordinate, normalized between 0 and 1, of the center of the person's head in the image for regression approach; and the resized 256×256 pixels image of the whole frame.

Outputs: For regression, the gaze target is represented as a (x, y) coordinate pair. For classification, the gaze target consists of a soft-label vector indicating which of the $N \times N$ grid cells contains the gaze target. In addition (for both regression and classification), the network also contains an "in"/"out" binary prediction of whether the gaze target is inside or outside the image.

Flow of architecture: The intuition behind the Merged Model is that two CNNs are trained to solve two subproblems in a fully end-to-end fashion with only the gaze location and the "in"/"out" label as supervision to the model: (1) The close-up face CNN (left pathway in Figure 4.2) implicitly estimates the head pose and the direction of the gaze of the subject in order to produce a heatmap (shown as `Reshape 16×16` in Figure 4.2) of where the person is looking. In the figure, the heatmap roughly shows a "cone" of possible gaze targets to the upper-left of the child's head. (2) The frame-image CNN (right pathway in Figure 4.2) identifies the salient objects in the image. This network has access to the entire original image but does not know the location of the subject. In the figure, the salient object heatmap highlights the teacher in the upper-left of the image. In [72], the authors showed that objects tend to emerge in the filter kernels of the deep layers of CNNs; therefore, we take a filter of the learnt representations at the end of the right pathway (shown as 3×3 `conv, 1` in Figure 4.2). This produces the heat map of salient objects in the original image. Heatmaps from both branches are merged by element-wise multiplication. After the merge, the two fully connected layers, separated by a dropout layer ($P = 0.3$), follows before branching off into two outputs. The loss for `Gaze Output` uses Euclidean loss for regression approach but uses Cross-entropy loss for classification approach. The loss for `In/Out Output` uses Cross-entropy loss for both regression and classification approach.

4.2.1 Training procedure

Data partitions: The 70 YouTube videos containing school classrooms were partitioned into training (12,430 gazes), validation (2,664 gazes), and testing (2,664 gazes) sets, such that none of the frames from any video was assigned to more than one set. Training data is augmented by flipping the original images and gazes left to right. This essentially doubles the training data size. The validation set was used for early stopping. The accuracy on the test set can be considered a performance estimate on faces that the network has never seen before.

Optimization: We used the following procedure for both the regression and classification formulations: We first performed transfer learning by initializing both CNNs with weights pre-trained on ImageNet [61]. We augmented the classroom images from our dataset by flipping the original images (frame image pathway) as well as the individually cropped face images, head locations and gaze locations (face pathway) left to right. We trained the final Merged Model first by freezing all the convolutional layers and training only the fully connected layers with RMSProp [28] (learning rate = 0.01, $\rho = 0.9$). Then all the previously frozen convolutional layers were unfrozen and the model was fine tuned with SGD with momentum (learning rate=0.0001, momentum=0.9). The model was trained until there is no improvement in validation loss.

4.3 Multi-task learning

Since the Merged Model predicts the location of the gaze in the image as well as "in"/"out", it is performing multiple tasks, and we can use multi-task learning (MTL) [13] for training. Merged model is minimizing two losses at the same time during training (Euclidean distance and cross entropy loss for regression and two cross entropy losses for classification) while sharing the same hidden layers as shown in Figure 4.2. Sharing same hidden layers to solve several tasks forces the model to find representations which capture all of the tasks and thus greatly reducing the risk of overfitting [5]. We found empirically that MTL helped to reduce overfitting and improve prediction accuracy, and we thus adopted the approach for training.

4.3.1 Effects of Multi-task learning

Effects of multi-task learning can be seen in Table 4.1. With multi-task learning, the model has higher training cross entropy losses (both for 8×8 output grid for gazes and In/Out gaze classification) compared to the models which minimizes only one of two losses. But the model with multi-task learning performs better on test set. All models have the same architecture, hyperparameters and are trained with early stopping with respect to the validation loss.

Table 4.1: Effects of multi-task learning

	Only grid output	Only In/Out output		Both grid output and In/Out output		
	CE Loss	CE Loss	AUC	CE Loss (Grid Output)	CE Loss (In/Out)	AUC (In/Out)
Training	3.27	0.32	0.63	3.39	0.33	0.60
Validation	3.49	0.47	0.60	3.58	0.44	0.58
Testing	3.59	0.46	0.59	**3.58**	**0.43**	**0.62**

Chapter 5

Who Are They Looking At?

In school classrooms, both students and teachers often look at other people's *faces*; the target of a student's gaze could be the face of the teacher who is teaching the class, or of another student with whom the student is collaborating (during group-based instruction) or socializing off-task. Identifying where each student is looking at can give information on where their attention is and their level of engagement. Moreover, the teachers themselves can vary the fixation of their gaze on different students at different points in time. They may notice important events such as when a student becomes confused or frustrated. They might also sometimes miss important dynamics between students, such as when one student within a group dominates the activity or acts inappropriately (e.g., bullying) towards another student. Knowing the gazes of students and teachers and identifying whether they are looking at one another can be a useful indicator of recognizing student-teacher interaction. We explore whether our deep neural network can pick up such gazes where people are looking at one another.

We can use the same neural network depicted in Figure 4.2 to predict *who* each person is looking at. This is especially useful in school classrooms, in which both students and teachers are often looking at other *people*, not just objects. Specifically, we use the *classification* approach to predict which of the 8×8 grid cells each person is gazing at. The face contained within that cell is then predicted to be target face of that person's gaze. We note that, depending on the grid size and the specific image, multiple faces might appear within the same cell and therefore, our accuracy calculation is only approximate. A principled approach to handle to this issue would be to distribute the probability mass output by the neural network among all the faces within that cell in proportion to the the size of each face. However, in this work, we simply assume that no grid cell contains more than 1 face.

5.1 Methodology

First, we computed the subset of all people in all image frames of our original YouTube dataset in which *all* annotators agreed that the person is looking at another *face* (not just another object somewhere in the image). Note that the labelers can still differ as to which particular face the person is looking at (Figure 5.1). By doing so, we obtained, 410 faces where all labelers agree that the person is looking at another face out of $17,759$ faces in our dataset. On the same data subset, we use the Merged Model to compute the softmax probabilities across all 8×8 grid cells of where each person was looking. From these probability outputs (for each person in each image), we remove every cell that does not contain any face (as determined by the face detector) and *normalize* the softmax probabilities only on the cells with faces. Step-by-step process of masking softmax output with face cells on 8×8 grid is depicted in Figure 5.2. In order to evaluate how well the our network is performing on determining which face a person is looking at, we took the top 1 face, top 2 faces and top 3 faces. The method of taking top-k faces is analogous to taking top-1 and top-5 error rates in evaluating image classifiers [38]. For top-1 face, we choose the grid cell with the highest probability as the face that the person is most likely to be gazing at as predicted by the deep neural network. For top-2 and top-3 faces, if any of the top-2 and top-3 faces predicted by the network is the actual face which is agreed by the majority of human labelers, the prediction is regarded as a correct prediction.

As baselines, we can consider that the average number of faces (detected by the face detector [30]) per image was 6.87 on test set; hence, the baseline guess rate is $1/6.87 \approx 0.146$ for the test set. Moreover, we can estimate human accuracy in a leave-one-labeler-out fashion: for each unique labeler, in the subset of the dataset where all labelers agree that a person being annotated is looking at another face, we compare the face that the current labeler chooses with the face which the majority of other labelers agree on. In this fashion, we compute the accuracy (% correct) of the l^{th} labeler w.r.t. the other $l - 1$ labelers. We then average across all labelers in our dataset. By doing so, we achieve the human level performance on determining whom the person is looking at in the classroom given that the person is looking at *a face*. Figure 5.1 shows some samples of human annotations where all labelers agree that the person being labeled is looking at *some* face. If the annotated gaze ends in one of the face bounding boxes, that gaze is considered to be looking at a face. Table 5.1 shows the probability of human labelers agreeing a person looking at another *specific* face.

In order to make equal comparison with Merged Model's predictions, which is done on 8×8 grid, human annotations are quantized to cells on 8×8 grid. The results for identifying whom the person is looking at annotated by humans and as predicted Merged Model on 8×8 grid are described in Section 6.4.

Table 5.1: Probability of humans agreeing that a person being labeled is looking at a particular face. Face Agreement represents the $n\%$ of labelers who label that a face is looking at another face.

Face Agreement	**100%**	**75%**	**50%**
Human	0.81	0.80	0.69

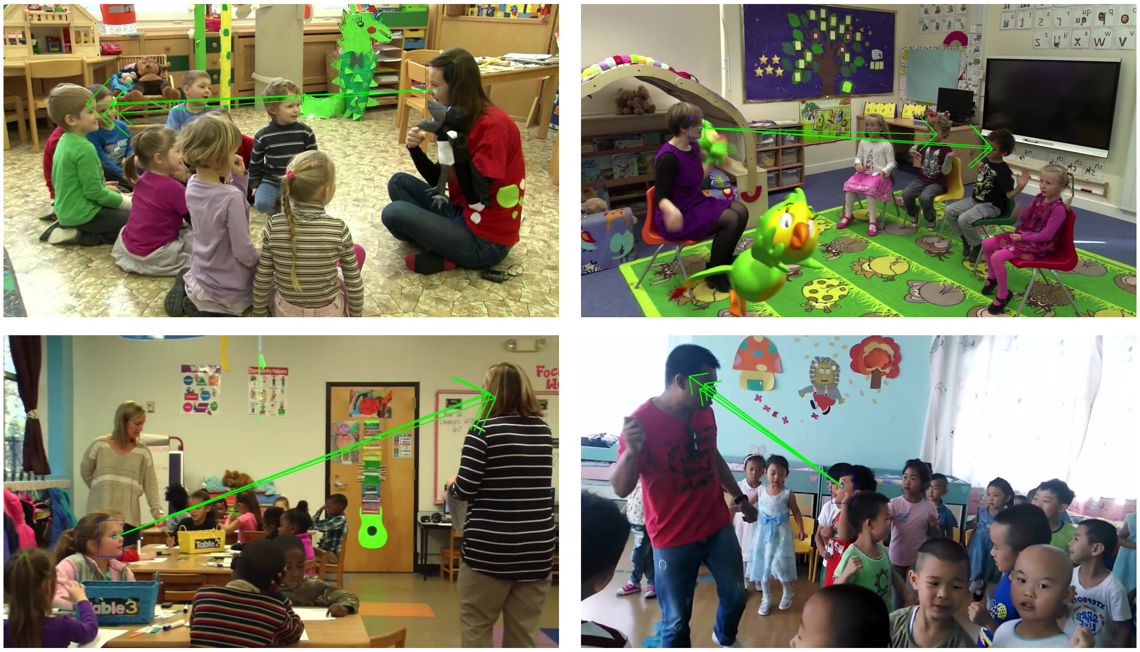


Figure 5.1: Students and teachers looking at faces of other students and teachers. **Top two figures**: annotators disagree on whom the face being annotated is looking at. **Bottom two figures:** annotators agree on whom the face being annotated is looking at. Images (in clockwise direction) taken from: `https://goo.gl/mQzXhu`, `https://goo.gl/F5jUCs`, `https://goo.gl/pb19gi`, `https://goo.gl/rCVudX`.

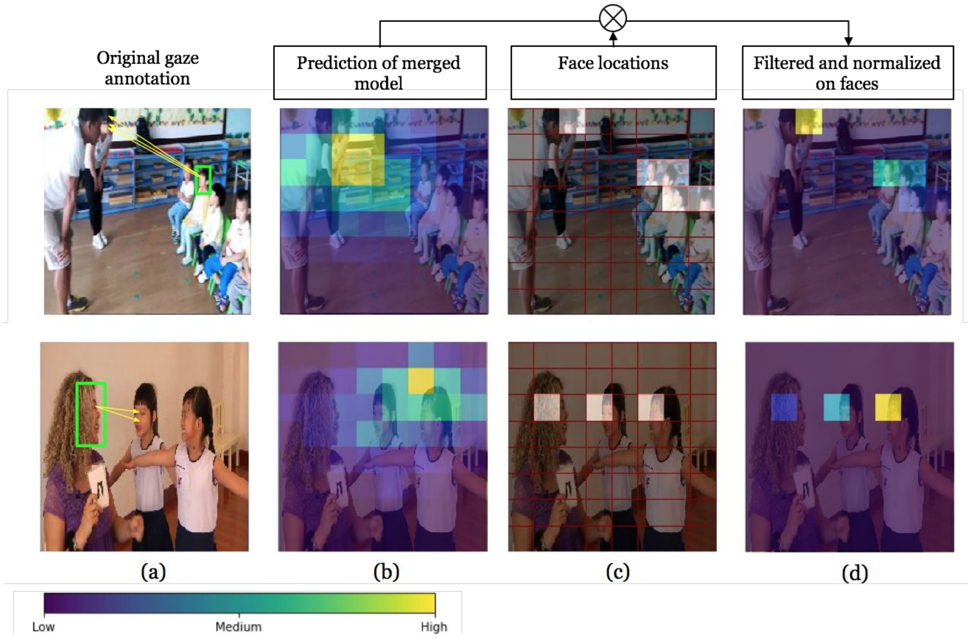


Figure 5.2: Step-by-step process of masking face cells for Merged Model outputs. (a) The original image with green box representing the person of interest and yellow line representing the human gaze annotation. (b) Softmax output of Merged Model on 8 × 8 grid. (c) Face cells on 8 × 8 grid. (d) Filtered and *normalized* softmax outputs which correspond to face cells.

Chapter 6

Evaluation and Results

In this chapter, we define several accuracy measurements which we use to measure the performance of our deep convolutional neural network for gaze following. In order to make a good assessment for any machine learning approach, we need to have several comparisons which give us an idea of how well our chosen machine learning approach performs with respect to other approaches. Therefore, we present several baselines which we use in this thesis work to measure the performance of our Merged Model. We treat human performance as our benchmark and since we have multiple human labelers annotating gaze locations in our dataset, we can obtain human level accuracy (details in Section 3.3.1). Since we approach the problem of gaze following either as a regression task or as a classification task (as described in Section 4.1), we report both the quantitative and qualitative results of both approaches. Finally we present the results of how well our Merged Model can predict whom the person is looking at given that the person is looking at some other face in the image.

6.1 Accuracy measurement

Accuracy is measured for predicting the gaze target of each person (identified automatically by a face detector [30]) in each extracted frame from each of the YouTube videos (see Section 3.1). For **classification** on 8×8 grid, we evaluate accuracy in terms of the cross entropy (CE) loss with respect to the label distribution induced by the 3 annotators per example. For **regression** to a (x, y) location, we use mean absolute error (MAE), mean Euclidean distance and mean angular error (between predicted gaze and ground-truth gaze) in degrees, where the ground-truth is defined as the *average* annotation over all other annotators for that face. In addition (for both regression and classification), we also used the Area Under the Receiver Operating Characteristics Curve (AUC) to evaluate the binary classification of whether the target is inside or outside the field-of-view.

6.2 Baseline comparison

When assessing accuracy of any neural network, it is important to establish the relevant baselines for comparison. For classification, we use a uniform distribution over all 8×8 grid cells – in other words a random guess in the whole image as to where the person is gazing. Alternatively, we can assume a center prior (motivated by [32]), consisting of the center 2×2 grid cells over the 8×8 grid. A variation on the center prior is to place a 2-D Gaussian – whose standard deviation σ is optimized directly on the *test set* for best possible accuracy – centered on the middle of the image, and assign probabilities to the 8×8 cells based on the Gaussian probability density function. For regression, we use a center prior corresponding to a random gaze location within the center 10% region of the image. We also compare to randomly selected points in the image.

Finally, as a much stronger baseline based on actually examining the image, we also compare to a Face-to-Gaze model consisting of a CNN that takes a cropped, close-up face image and location of head in the image as inputs, and predicts the location of the gaze in the image as well as "in"/"out" – this is the left pathway of Figure 4.2. Comparing with this baseline helps us understand how much the saliency pathway improves performance.

6.3 Results: Predicting Gaze Locations

Accuracy results on test images of the Merged Model compared to the baselines are shown in in Table 6.1 (for regression) and Table 6.2 (for classification). Our Merged Model achieves mean Euclidean distance of 69.82 (in pixels) on 256×256 pixel image (for regression) and cross entropy loss of 3.58 on 8×8 grid (for classification) for gaze locations. These numbers are better than for the random gaze, center prior, or center Gaussian baselines. There is a significance difference in accuracy between Merged Model and Face-To-Gaze for gaze location predictions as (x, y) coordinates ($t(14659) = -8.92, p = 4.8^{-19}$, 2-tailed) as well as for gaze cell prediction on 8×8 grid ($t(5311) = 7.89, p = 3.7^{-15}$, 2-tailed).

For comparison, human labelers exhibited a mean Euclidean distance of only 41.04, which is a bit more than half the error of the Merged Model, indicating that the machine's accuracy still has much room for improvement.

One notable fact is that the **Face-to-Gaze** model's performance is very similar to the Merged Model's performance. This suggests that our Merged Model is predicting gaze locations mainly by using the head pose and gaze pathway of the subject and less on the salient objects in the image. One possible explanation is that our dataset does not contain enough variety of classroom environments for the model to learn how to identify salient objects in classroom images.

For classifying whether the gazes end inside or outside the image, the Merged Model achieved an AUC of 0.62, whereas humans scored 0.70 on the same task. The relatively low human accuracy suggests that detecting whether a person is looking inside or outside the image is quite challenging in the classroom images.

Figure 6.1 shows qualitative results of some of the gaze predictions (represented by thick yellow arrows) by Merged Model when using regression approach (i.e. predicting gaze locations as (x, y) coordinates in the image). It can be seen that the model makes decent predictions on general direction of gazes but sometimes misses the end-points on salient objects in the scene. In first row of Figure 6.1, three girls in the middle are looking at the man's hands but the gaze predictions end before the hand. This suggests that the Merged Model is not picking up salient objects that the people in the image might be looking at well enough to be able to extend gaze predictions to end on salient objects.

Qualitative results for predicting gaze cell on 8×8 grid is shown in Figure 6.2. Right column of Figure 6.2 represents softmax output distributions of gaze regions as predicted by the Merged Model for people in green bounding boxes. These results suggest that the Merged Model is able to predict the region of gaze by assigning more probability (yellower cells) to cells which correspond to the location in the image where the person is more likely to be looking at than those cells which correspond to locations where the gaze of that person is less likely to end (bluer cells).

6.4 Results: Who Are They Looking At?

The results of whom the person is looking at as predicted by the Merged Model on test set is shown in Table 6.3. The results indicate that the Merged Model can predict the face target of people's eye gazes with substantially higher accuracy than just randomly guessing among all grid cells in the image containing faces. To put these results in context: if each classroom image contains 6.87 faces on average (as reported in Section 5.1), then the probability of 0.79 for $k = 3$ suggests that an automated gaze following system can usually determine at least which *group* of students a teacher is looking at. Interestingly, the accuracy of the Merged Model is close to that of human labelers when top 3 predicted faces are considered. Since Merged Model's performance is measured on predicted gaze cell on 8×8 grid, gaze annotations of humans are converted to a cell on 8×8 grid to calculate the human performance on whether all human annotators agree on a *specific* face.

Table 6.1: Regression accuracy of the Merged Model for predicting the (x, y) location (within a 256×256 pixel image). Accuracy is compared to human annotators and three baselines. **Random Gaze** is random points over the whole image and **Center Region** is random points within the center 10% of the image. **Face-to-Gaze** convolutional neural network regresses to (x, y) locations of gazes using only head locations and cropped, close-up face images as inputs.

	MAE (pixels)	Mean Euclidean Distance (pixels)	Mean Absolute Angular Error	AUC for In/Out
Random Gaze	79.74	124.15	67.24°	-
Center Region	52.76	82.11	48.36°	-
Face-to-Gaze	45.74	71.53	39.91°	0.54
Merged Model	**44.49**	**69.82**	**38.30°**	**0.62**
Human	25.91	41.04	18.38°	0.70

Table 6.2: Classification results on 8×8 grid of the Merged Model compared to several baselines **Center Gaze (Center 4 cells)** is a uniform distribution over 4 center cells in the image and **Center Gaussian** is a centered mean 2D gaussian distribution with standard deviation which gives the best result on the test set. **Uniform Gaze** is a uniform distribution over the whole 8×8 grid. **Face-to-Gaze** convolutional neural network predicts the location of gaze on 8×8 grid using only head locations (on 8×8 grid) and cropped, close-up face images as inputs.

	Cross Entropy Loss (Grid Output)	AUC for In/Out
Center Gaze (Center 4 cells)	15.80	-
Center Gaussian	4.06	-
Uniform Gaze	4.16	-
Face-to-Gaze	3.75	0.54
Merged Model	**3.58**	**0.62**

Table 6.3: Probability of the Merged Model correctly identifying which face a person is looking at on 8×8 grid.

Top k faces	$k = 1$	$k = 2$	$k = 3$
Random Face	0.15	0.30	0.45
Merged Model	**0.47**	**0.65**	**0.79**
Human	0.82		

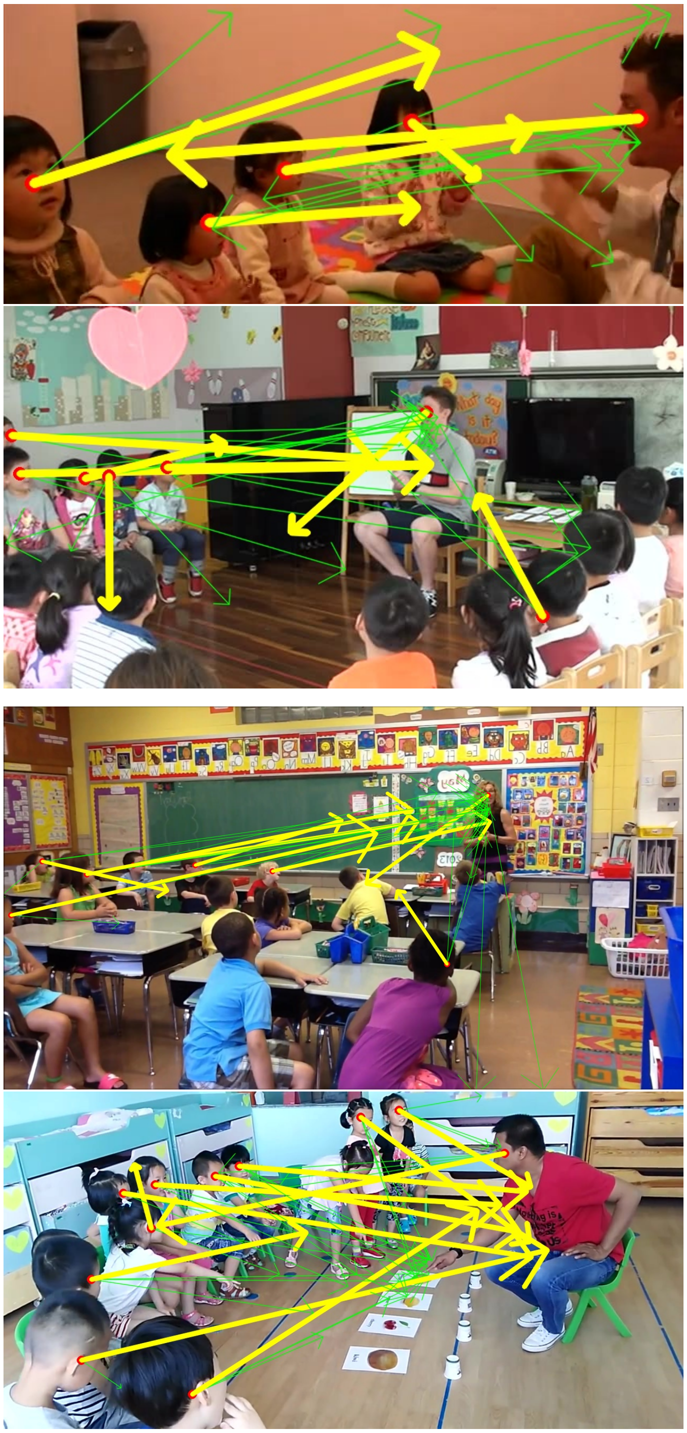


Figure 6.1: Handpicked qualitative results of gaze predictions by Merged Model using regression approach on test set. Thin green arrows are ground truth annotations. Since there are multiple gaze annotations for each individual, there are multiple green arrows for each individual. Thick yellow arrows are predictions by Merged Model. Images (top to bottom) taken from: `https://goo.gl/xUdYbC`, `https://goo.gl/FBSA57`, `https://goo.gl/v931J6`, `https://goo.gl/pcwQ5P`

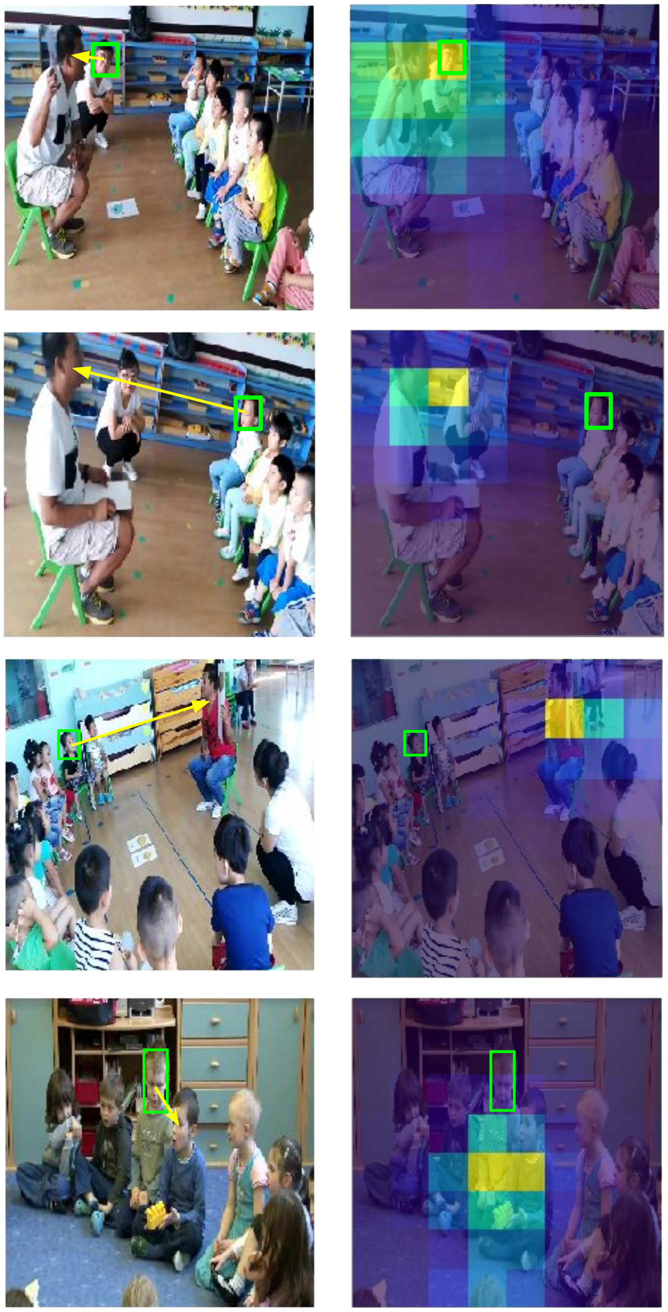


Figure 6.2: Handpicked qualitative results of gaze prediction by Merged Model using classification approach on test set. **Left Column:** Input images resized to 256×256 pixels. Green bounding boxes are person of whose gaze are being predicted and yellow lines are human labeled gaze annotations. **Right Column:** Softmax output distribution on 8×8 grid predicted by Merged Model for corresponding images and person of interest (indicated by green bounding box). Softmax output distribution forms a heatmap of the region of gaze by the person in green bounding box.

Chapter 7

Conclusion and Future Work

In this thesis work, we have built a deep neural network, based on the approach by [56] that analyzes 2-D images of classrooms to predict gaze locations of students and teachers. The results indicate that our network can estimate the gaze target locations with substantially higher than chance and better than several other baselines. But when compared to human level performance, our deep neural network still has room for improvement. The same architecture can be used to identify *who* each person is looking at more accurately than random guessing. We also showed that multi-task learning, not explored by [56], helps our model regularize better. Future researchers can explore building gaze predicting networks which incorporate other tasks as well (e.g. identifying whether the gaze end point is on an object of interest or not).

Gaze prediction of people in images have been tackled by both traditional computer vision approaches and deep neural networks but most of the prior work has been done on general computer vision research datasets. In this thesis work, we take a step towards building gaze prediction system *specifically* for classrooms by: 1) collecting and annotating gaze dataset of multiple students and teachers from recorded classroom observation videos; 2) building a deep neural network to predict eye gazes of students and teachers using both regression and classification approach; 3) Identifying who is looking at whom in classrooms; 4) Identifying whether gazes end inside the image or extend outside the image.

In the course of this thesis work, one difficulty we encountered is considering the best way to annotate gaze in images due to the challenge of capturing depth information from 2-D images. Not only that, the task of crowd-source labelling 2-D classroom images which have cluttered backgrounds and contain multiple students and teachers presents as a challenge for annotators as well. This is observed in our inter-labeler reliability tests in Section 3.3. We hope that future researchers can make more informed decisions on better ways of collecting gaze data on 2-D images based on our work.

One of the main intentions of our gaze prediction system is to provide gaze information of multiple individuals which can be used in downstream processing of identifying attentions of teachers and students. We hope that the gaze predicting deep neural network built in this thesis work can serve as a building block in a more comprehensive automated classroom assessment system.

7.1 Future Work

There are several venues that can be explored from this thesis work: (1) Collecting more gaze annotations and improving the inter-labeler reliability. Classroom gaze dataset collected in this thesis work is somewhat limited in number. Therefore, it would be worthwhile to collect more data from different classroom settings to train a better gaze following deep neural network. One crucial next step can be exploring different ways to help labelers have less uncertainties when it comes to annotating gaze in 2D images. (2) Estimating gaze jointly rather than separately. Since multiple people often look at the same person (e.g., the teacher) in school classrooms, it would be a good next step to investigate whether gaze location accuracy of each person can be improved by estimating the gaze targets of all classroom participants *jointly* rather than separately. (3) Using object detectors to identify various objects and people in classrooms. This can help annotation process in a way such that labelers can draw their gazes *into* the boxes which bound people and objects. By collecting this meta information, we can have a richer set of data which can facilitate training gaze following deep neural networks. (4) Finally, given an improved eye gaze following system, future work can explore how automatic gaze estimates can be used to predict specific aspects of classroom observation protocols; for instance, the *positive climate* dimension of the CLASS is based explicitly (in part) on whether the teacher looks at his/her students [53].

Bibliography

[1] Aung, A. M., and Whitehill, J. R. Harnessing label uncertainty to improve modeling: An application to student engagement recognition. In *IEEE Automatic Face & Gesture Recognition* (2018).

[2] Baldwin, D. A. Understanding the link between joint attention and language. *Joint attention: Its origins and role in development* (1995), 131–158.

[3] Baron-Cohen, S., Campbell, R., Karmiloff-Smith, A., Grant, J., and Walker, J. Are children with autism blind to the mentalistic significance of the eyes? *British Journal of Developmental Psychology 13*, 4 (1995), 379–398.

[4] Baum, E. B., and Haussler, D. What size net gives valid generalization? In *Advances in neural information processing systems* (1989), pp. 81–90.

[5] Baxter, J. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine learning 28*, 1 (1997).

[6] Bennett, S., and Barp, D. Peer observation–a case for doing it online. *Teaching in Higher Education 13*, 5 (2008), 559–570.

[7] Bock, S. W., Dicke, P., and Thier, P. How precise is gaze following in humans? *Vision research 48*, 7 (2008), 946–957.

[8] Borji, A., Cheng, M.-M., Hou, Q., Jiang, H., and Li, J. Salient object detection: A survey. *arXiv preprint arXiv:1411.5878* (2014).

[9] Borji, A., Parks, D., and Itti, L. Complementary effects of gaze direction and early saliency in guiding fixations during free viewing. *Journal of vision 14*, 13 (2014).

[10] Bosch, N., D'Mello, S., Baker, R., Ocumpaugh, J., Shute, V., Ventura, M., Wang, L., and Zhao, W. Automatic detection of learning-centered affective states in the wild. In *International conference on intelligent user interfaces* (2015).

[11] Brooks, R., and Meltzoff, A. N. Gaze following: A mechanism for building social connections between infants and adults. *In Mechanisms of social connection: from brain to group* (2014).

[12] Carnevali, P., and Patarnello, S. Exhaustive thermodynamical analysis of boolean learning networks. *EPL (Europhysics Letters) 4*, 10 (1987), 1199.

[13] Caruana, R. Multitask learning: A knowledge-based source of inductive bias. In *International Conference on Machine Learning* (1993).

[14] Curby, T. W., Johnson, P., Mashburn, A. J., and Carlis, L. Live versus video observations: Comparing the reliability and validity of two methods of assessing classroom quality. *Journal of Psychoeducational Assessment 34*, 8 (2016), 765–781.

[15] Dalal, N., and Triggs, B. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (2005), vol. 1, IEEE, pp. 886–893.

[16] Deng, T., Yang, K., Li, Y., and Yan, H. Where does the driver look? top-down-based saliency detection in a traffic driving environment. *IEEE Transactions on Intelligent Transportation Systems 17*, 7 (2016).

[17] Denker, J., Schwartz, D., Wittner, B., Solla, S., Howard, R., Jackel, L., and Hopfield, J. Large automatic learning, rule extraction, and generalization. *Complex systems 1*, 5 (1987), 877–922.

[18] D'Mello, S. K., Olney, A. M., Blanchard, N., Samei, B., Sun, X., Ward, B., and Kelly, S. Multimodal capture of teacher-student interactions for automated dialogic analysis in live classrooms. In *ACM international conference on multimodal interaction* (2015).

[19] Donnelly, P. J., Blanchard, N., Samei, B., Olney, A. M., Sun, X., Ward, B., Kelly, S., Nystrand, M., and D'Mello, S. K. Multi-sensor modeling of teacher instructional segments in live classrooms. In *ACM international conference on multimodal interaction* (2016).

[20] Emery, N. J. The eyes have it: the neuroethology, function and evolution of social gaze. *Neuroscience & Biobehavioral Reviews 24*, 6 (2000).

[21] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. The pascal visual object classes (voc) challenge. *International journal of computer vision 88*, 2 (2010).

[22] Fathi, A., Hodgins, J. K., and Rehg, J. M. Social interactions: A first-person perspective. In *Computer Vision and Pattern Recognition* (2012).

[23] Fathi, A., Li, Y., and Rehg, J. M. Learning to recognize daily actions using gaze. In *European Conference on Computer Vision* (2012), Springer, pp. 314–327.

[24] Flom, R., Lee, K., and Muir, D. *Gaze-following: Its development and significance.* Psychology Press, 2017.

[25] Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. *Deep learning*, vol. 1. MIT press Cambridge, 2016.

[26] Grafsgaard, J., Wiggins, J. B., Boyer, K. E., Wiebe, E. N., and Lester, J. Automatically recognizing facial expression: Predicting engagement and frustration. In *Educational Data Mining* (2013).

[27] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 770–778.

[28] Hinton, G. Rmsprop: Divide the gradient by a running average of its recent magnitude.

[29] Hoffman, M. W., Grimes, D. B., Shon, A. P., and Rao, R. P. A probabilistic model of gaze imitation and shared attention. *Neural Networks 19*, 3 (2006), 299–310.

[30] Jiang, H., and Learned-Miller, E. Face detection with the faster r-cnn. In *IEEE Automatic Face & Gesture Recognition* (2017).

[31] Joe, J. N., Tocci, C. M., Holtzman, S. L., and Williams, J. C. Foundations of observation. *Princeton, NJ: Educational Testing Service* (2013).

[32] Judd, T., Ehinger, K., Durand, F., and Torralba, A. Learning to predict where humans look. In *International Conference on Computer Vision* (2009).

[33] Kane, T. J., McCaffrey, D. F., Miller, T., and Staiger, D. O. Have we identified effective teachers? validating measures of effective teaching using random assignment. In *Research Paper. MET Project. Bill & Melinda Gates Foundation* (2013).

[34] Kapoor, A., Burleson, W., and Picard, R. W. Automatic prediction of frustration. *International journal of human-computer studies 65*, 8 (2007).

[35] Kleinke, C. L. Gaze and eye contact: a research review. *Psychological bulletin 100*, 1 (1986), 78.

[36] Kontos, S., and Wilcox-Herzog, A. Teachers' interactions with children: Why are they so important? research in review. *Young Children 52*, 2 (1997).

[37] Krafka, K., Khosla, A., Kellnhofer, P., Kannan, H., Bhandarkar, S., Matusik, W., and Torralba, A. Eye tracking for everyone. In *Computer Vision and Pattern Recognition* (2016).

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

[39] Landis, J. R., and Koch, G. G. The measurement of observer agreement for categorical data. *biometrics* (1977), 159–174.

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

[41] LeCun, Y., et al. Generalization and network design strategies. *Connectionism in perspective* (1989), 143–155.

[42] Lin, M., Chen, Q., and Yan, S. Network in network. *arXiv preprint arXiv:1312.4400* (2013).

[43] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European conference on computer vision* (2014).

[44] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision 60*, 2 (2004), 91–110.

[45] Marín-Jiménez, M. J., Zisserman, A., Eichner, M., and Ferrari, V. Detecting people looking at each other in videos. *International Journal of Computer Vision 106*, 3 (2014).

[46] Mashburn, A. J., Pianta, R. C., Hamre, B. K., Downer, J. T., Barbarin, O. A., Bryant, D., Burchinal, M., Early, D. M., and Howes, C. Measures of classroom quality in prekindergarten and childrens development of academic, language, and social skills. *Child development 79*, 3 (2008), 732–749.

[47] McMahon, T., Barrett, T., and O'Neill, G. Using observation of teaching to improve quality: Finding your way through the muddle of competing conceptions, confusion of practice and mutually exclusive intentions. *Teaching in Higher Education 12*, 4 (2007), 499–511.

[48] Mukherjee, S. S., and Robertson, N. M. Deep head pose: Gaze-direction estimation in multimodal video. *IEEE Transactions on Multimedia 17*, 11 (2015).

[49] Nair, V., and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)* (2010), pp. 807–814.

[50] Oquab, M., Bottou, L., Laptev, I., and Sivic, J. Learning and transferring mid-level image representations using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on* (2014), IEEE, pp. 1717–1724.

[51] Parks, D., Borji, A., and Itti, L. Augmented saliency model using automatic 3d head pose detection and learned gaze following in natural scenes. *Vision research 116* (2015), 113–126.

[52] Pianta, R. C., Hamre, B. K., and Allen, J. P. Teacher-student relationships and engagement: Conceptualizing, measuring, and improving the capacity of classroom interactions. In *Handbook of research on student engagement*. Springer, 2012, pp. 365–386.

[53] Pianta, R. C., La Paro, K. M., and Hamre, B. K. *Classroom Assessment Scoring System^{TM}: Manual K-3.* Paul H Brookes Publishing, 2008.

[54] Premack, D., and Woodruff, G. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences 1*, 4 (1978), 515–526.

[55] Ramakrishnan, A., and Whitehill, J. Youtube pre-school dataset, 2017.

[56] Recasens, A., Khosla, A., Vondrick, C., and Torralba, A. Where are they looking? In *Advances in Neural Information Processing Systems* (2015).

[57] Recasens, A., Vondrick, C., Khosla, A., and Torralba, A. Following gaze in video. In *Computer Vision and Pattern Recognition* (2017).

[58] Rosenblatt, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review 65*, 6 (1958), 386.

[59] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision 115*, 3 (2015).

[60] Shortland, S. Peer observation: A tool for staff development or compliance? *Journal of further and higher education 28*, 2 (2004), 219–228.

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

[62] Symons, L. A., Hains, S. M., and Muir, D. W. Look at me: Five-month-old infants' sensitivity to very small deviations in eye-gaze during social interactions. *Infant Behavior and Development 21*, 3 (1998), 531–536.

[63] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., et al. Going deeper with convolutions. Cvpr.

[64] Tensorflow. Deepdreaming with tensorflow.

[65] Tishby, N., Levin, E., and Solla, S. A. Consistent inference of probabilities in layered networks: Predictions and generalization. In *IJCNN International Joint Conference on Neural Networks* (1989), vol. 2, IEEE New York, pp. 403–409.

[66] Triesch, J., Teuscher, C., Deák, G. O., and Carlson, E. Gaze following: why (not) learn it? *Developmental science 9*, 2 (2006), 125–147.

[67] Wang, Z., Pan, X., Miller, K. F., and Cortina, K. S. Automatic classification of activities in classroom discourse. *Computers & Education 78* (2014).

[68] Wotton, R. S. Natural history, creation and religious conflicts, Jan 1970.

[69] Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR)* (2010).

[70] Yao, B., Jiang, X., Khosla, A., Lin, A. L., Guibas, L., and Fei-Fei, L. Human action recognition by learning bases of action attributes and parts. In *International Conference on Computer Vision* (2011).

[71] Zain, N. H. M., Razak, F. H. A., Jaafar, A., and Zulkipli, M. F. Eye tracking in educational games environment: evaluating user interface design through eye tracking patterns. In *International Visual Informatics Conference* (2011).

[72] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856* (2014).

[73] Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems* (2014).