Convolutional Neural Network Implementation for Eye-Gaze Estimation on Low-Quality Consumer Imaging Systems

Joseph Lemley, *Student Member, IEEE*, Anuradha Kar, *Student Member, IEEE*, Alexandru Drimbarean, *Member, IEEE*, and Peter Corcoran, *Fellow, IEEE*

***Abstract*—Accurate and efficient eye gaze estimation is important for emerging consumer electronic systems, such as driver monitoring systems and novel user interfaces. Such systems are required to operate reliably in difficult, unconstrained environments with low power consumption and at minimal cost. In this paper, a new hardware friendly, convolutional neural network (CNN) model with minimal computational requirements is introduced and assessed for efficient appearance-based gaze estimation. The model is tested and compared against existing appearance-based CNN approaches, achieving better eye gaze accuracy with significantly fewer computational requirements.**

***Index Terms*—Eye gaze, neural networks, deep learning.**

I. Introduction

The potential of eye gaze tracking and gaze-based human computer interactions in modern consumer devices is currently an active topic for exploration. Eye gaze has been used to derive human behavioral cues, as an input modality and for achieving immersive user experiences in virtual and augmented reality systems. However, applications of gaze in consumer devices operating in real world conditions face tough challenges in terms of accuracy and reliability.

A. Gaze Tracking in Consumer Devices

After decades of research on desktop-based gaze estimation techniques, the focus has recently shifted to building eye gaze applications for dynamic platforms such as driver monitoring systems [1] and handheld devices [2]. For an automobile driver, eye-based cues such as levels of gaze variation, speed of eyelid movements and eye closure can be indicative of a driver's cognitive state. These can be useful inputs for intelligent vehicles to understand driver attentiveness levels, lane change intent, and vehicle control in the presence of obstacles to avoid accidents [3]. Handheld devices like smartphones and tablets form unique platforms for gaze tracking applications wherein gaze may be used as an input modality for device control, activating safety features and novel user interface (UI) designs [4].

The most challenging aspect of these modern gaze applications includes operation under dynamic user conditions and unconstrained environments. Further requirements for implementing a consumer-grade gaze tracking system include real-time high-accuracy operation, minimal or no calibration, and robustness to user head movements and varied lighting conditions. Therefore accurate and reliable gaze tracking typically demands high quality cameras and special equipment like narrow angle lenses, external illumination, and stereo setups for capturing eye region features with sufficient details. As a result, gaze estimation systems frequently become costly with complicated setups, which are unsuitable for generic and consumer applications.

Therefore a major challenge of gaze based consumer electronics design involves maximizing system performance while reducing costs and system complexities.

B. Deep Learning for Eye Gaze

In this paper, we introduce a calibration-free method for appearance-based gaze estimation that is suitable for consumer applications and low cost hardware with real time requirements, using a Convolutional Neural Network (CNN). CNNs were popularized by LeCun *et al.* [5], who used them successfully for handwritten digit classification. These networks are inspired by the organization of the visual cortex and allow spatial information to be more efficiently learned. Convolutional Neural Networks can be used on input with any number of dimensions, but due to their success in pictures, are most popularly implemented for 2D input plus color channels. Other popular types of CNNs include 1D CNNs, which are commonly used for time series, and 3D CNNs, which can be used for volumetric data or time series data where the third dimension represents either spatial frames or temporal frames [6]. Although CNNs have become ubiquitous for most computer vision tasks, they have yet to become popular for eye gaze estimation.

Manuscript received June 7, 2018; revised September 9, 2018, December 15, 2018, and January 27, 2019; accepted February 13, 2019. Date of publication February 15, 2019; date of current version April 23, 2019. This work was supported in part by the Science Foundation Ireland (SFI) Strategic Partnership Program by SFI and Fotonation Ltd., on Next Generation Imaging for Smartphone and Embedded Platforms under Project 13/SPP/I2868, and in part by the Irish Research Council Employment-Based Programme Award under Project EBPPG/2016/280. *(Corresponding author: Joseph Lemley.)*

J. Lemley, A. Kar, and P. Corcoran are with the Department of Electrical and Electronic Engineering, National University of Ireland Galway, Galway, Ireland (e-mail: j.lemley2@nuigalway.ie; a.kar2@nuigalway.ie; peter.corcoran@nuigalway.ie).

A. Drimbarean is with Vice President of Advanced Research, Fotonation Ltd., Galway, Ireland (e-mail: alexandru.drimbarean@xperi.com).

Digital Object Identifier 10.1109/TCE.2019.2899869

Neural network implementations are particularly important for embedded and low-power consumer imaging systems because hardware based CNNs can provide significant improvement in power efficiencies over CPU/GPU based solutions. A useful comparison of the performance gains of Tensor Processing Units (TPU), a custom Application Specific Integrated Circuit (ASIC) running deep neural networks, with conventional GPU arrays is given in [7]. The TPU is on average about 15X - 30X faster than its contemporary GPU or CPU, with TOPS/Watt about 30X - 80X higher depending on the percent usage of the TPU. While the use case described in [7] is for applications running in a cloud datacenter, equivalent embedded and mobile ASICs are expected to emerge in the next 1-2 years with the capability to operate multiple, parallel CNN-networks with equivalent levels of power efficiency to the TPU.

C. Contributions of This Work

From the perspective of developing a deep learning model for gaze estimation, the task can either be considered as a regression task or a classification task. Although both are useful, regression provides the greatest predictive flexibility and thus this paper treats the eye gaze estimation task as a regression problem with the goal of finding a gaze angle (ϕ, θ) that corresponds with a low resolution eye image such as one taken from a distance with a simple RGB webcam mounted on a dashboard.

In this paper, a hardware optimized network is implemented with demonstrated suitability for deployment on such consumer devices in terms of memory requirements and speed. This network achieves superior accuracy using a dual channel input technique when compared to other state-of-the-art CNN-based gaze tracking methods for unconstrained, low resolution eye tracking.

II. Related Work

In this section a review of conventional gaze tracking techniques, studies on using low resolution data, and the application of deep learning in gaze estimation are discussed.

A. Contemporary Methods for Eye Gaze Estimation

Gaze tracking algorithms can be broadly classified into model-based methods and appearance-based methods [8]. Appearance-based methods operate directly on the eye images. Model-based methods include 2D and 3D models that use near infrared (NIR) illumination to create corneal reflections to estimate the gaze vector.

Contemporary research on gaze tracking measures accuracy in a wide variety of ways [9]–[12]. It is the view of the authors that angular resolution is most reliable and in this work it is used as the metric of accuracy for the proposed algorithm and results are only compared directly with other works that employ the same metric.

These require polynomial or geometric approximations of the human eye to obtain the gaze direction or the point of gaze. Appearance-based methods use eye region images to extract content information such as local features, shape, and texture of eye regions, to estimate gaze direction.

1) 2D Models: 2D models utilize polynomial transformation functions for mapping the gaze vector (vector between pupil center and corneal glint) to corresponding gaze coordinates on a device screen. A number of related works discuss the use of artificial neural networks to perform this mapping [13]–[15]. Early papers on this topic proposed support vector machines (SVM) [16] and non geometric methods [17]. Although the approach proposed in this paper differs, we share a similar goal: head pose invariant gaze tracking without a geometric model while treating the eye gaze estimation task as a 2D regression problem.

2) 3D Models: 3D model-based methods typically use a geometrical model of the human eye to estimate the center of the cornea, and the optical and visual axes of the eye [18]–[21]. Gaze coordinates are estimated as points of intersection of the visual axes with the scene. These methods achieve high accuracy (1 degree) but require elaborate system setups and knowledge about geometric relations between system components like LEDs, monitors and cameras.

Although such methods achieve high degrees of accuracy, they are not suitable for the use case investigated in this paper because they require high resolution images of the eye and precise geometric measurements. Such models represent current state-of-the-art in AR/VR systems [22].

3) Appearance-Based Methods: Appearance-based methods utilize cropped eye images of a subject gazing at known locations to generate gaze point coordinates. A variety of machine learning methods, using the eye images as input, have been suggested for this task. For the interested reader there is a detailed discussion of appearence based models by Kar and Corcoran [9]. Recently, appearance-based methods implemented using deep learning (DL) and convolutional neural network (CNN) approaches have gained momentum. In contrast the other methods discussed in this subsection, CNN methods, learn the features directly from the data rather than employing a preliminary feature detection step and are described in detail in Section II-C.

B. Gaze Estimation From Low Resolution Images

To facilitate gaze tracking in everyday settings, the use of cheap, compact and easy-to-integrate webcams is common but results in poor gaze estimation accuracy. Low resolution images have strong noise effects [23], and distortions in the eye region contours and features become indistinguishable under varying illumination levels, user distance, and movements.

C. Eye Gaze Estimation Using CNN's

Deep learning (DL) techniques have been successfully used in challenging conditions such as those with variable illumination, unconstrained backgrounds and free head motion. For example George and Routray [24] describe a calibration-free real-time CNN-based framework for gaze classification. Two CNNs, for the left and right eyes, are then trained independently to classify the gaze in seven directions.

Zhang *et al.* [25] describes a novel appearance-based gaze estimation method in which a CNN utilizes the full face image as input with spatial weights on the feature maps to suppress or enhance information in different facial regions. It achieves high accuracy and robust performance under varied illumination and extreme head poses. Deng and Zhu [26] achieves head pose invariant, 3D gaze tracking using two separate head pose and eye movement models with two CNNs, connected via a gaze transform layer. Finally, Zhang *et al.* [27] builds a CNN to learn the mapping between 2D head angle, eye image and gaze angle (output) using a small LeNet-inspired CNN. For testing, an extensive database is built with more than 200,000 images under variable illumination levels and eye appearances. This database (called MPII Gaze) is also used in this work and its further details are provided in the next section.

Several of the CNN-based works are specifically targeted towards gaze tracking in consumer/handheld devices [28], [29]. Krafka *et al.* [28] presents a CNN-based real time, calibration-free gaze estimation algorithm. It is trained using a large and diverse dataset of eye images taken under variable lighting, head pose, and backgrounds captured from users through a smartphone app. Inputs to their CNN model include eye and face images. The location of the faces in the images are obtained through a face grid, which is used to infer relative eye and head poses. Park and Kim [29] present a calibration-free method using Deep Belief Networks which classify gaze into a grid of nine gaze locations under various head-poses and viewing directions. Zhang *et al.* [30] developed a nine directional CNN-based gaze classifier for a screen typing application, robust to false detections, blinks, and saccades (rapid, abrupt changes in fixation).

D. Related Work Utilizing the MPII Gaze Dataset

Models for eye gaze estimation often have radically different errors on different datasets and this can make comparisons between claimed accuracies meaningless without knowing the accuracy on a common dataset. In this subsection, works that use the same dataset used in this paper are outlined to facilitate such comparison.

The MPII Gaze dataset [27] is large and challenging, containing images collected under a wide range of realistic scenarios, such as varied illumination levels, eye appearances and head poses. Use of MPII gaze dataset for training and testing gaze estimation algorithms can be found in their own papers [25], [27] and also in works by Wu *et al.* [31], Iyer and Ramasangu [32], and Nie *et al.* [33].

Zhang *et al.* [27], which introduces the MPII Gaze dataset, uses a multimodal CNN for gaze estimation and reports a cross dataset test error of 6.1 degrees. Zhang *et al.* [25] uses full face (instead of eye only and multi-region) images with a CNN and achieves a person independent error of 4.8 degrees on MPII Gaze while being robust to illumination variations and extreme head poses. Wu *et al.*, tracks gaze location using a CNN, and tests cross-subject performance on MPII Gaze in addition to that authors' own dataset [34]–[36].

Works by Wood *et al.* [34], [35], and Weidenbacher *et al.* [37] describe the utilization of the MPII Gaze dataset for comparing face models and other synthetic datasets.

Wood *et al.* [34] presents a method for synthesizing a large set of eye region images with a generative 3D eye region model. Then a gaze estimation method using the k-Nearest-Neighbour algorithm) is tested on the synthetic data and the MPII Gaze dataset to achieve an error of 9.95 and 9.58 degrees respectively. The same authors describe another method for synthetic, labelled photo-realistic eye region image creation using head scan geometry [35]. The generated dataset, along with MPII gaze, is used to test and compare the accuracy of a CNN based gaze estimation method. The CNNs are then tested on the MPII gaze dataset to achieve an error of 7.8 degrees.

E. Discussion

The previous sections summarized relevant work in the field of eye gaze estimation using regression, 3D models, CNNs, and low quality images. The purpose of the detailed review on these methods is to provide the reader with an understanding of the motivation and significance of the work described in this paper. While many papers have been written on conventional polynomial regression-based techniques for gaze estimation, these typically suffer from inaccuracy resulting from head movements and are not as accurate as 3D model-based methods. 3D model-based methods, on the other hand require complicated calibration, arrangement of the physical space, and intensive model calculations, which are often not suitable for implementation in consumer electronic devices. While CNNs have recently been used for gaze estimation, their accuracy is often insufficient and most implementations require powerful and expensive hardware such as graphic processing units (GPUs) due to large network sizes.

The work described in this paper builds on these techniques to achieve superior accuracy while using consumer grade hardware and inexpensive simple setups, which, as can be seen from the literature review, is an essential but relatively new direction of development in eye gaze research.

III. Methods

The CNN-based gaze estimation methods in this work were evaluated on state-of-the art graphic processing units using python 2.7 and caffe 1.0 with accuracy and Euclidean loss layers modified to calculate angle difference in radians. Person-exclusive, leave-one-out cross-validation was used in all experiments.

In eye gaze tracking literature it is common to use the word "accuracy" and "error" interchangeably and this can sometimes cause confusion. For this reason we use the word "error" in any case where the meaning could be unclear. All angles are reported in degrees. Although most deep learning papers will use just one training, testing, and validation set, such methods are not recommended in cases where there are few individuals in the dataset. For this reason we use a "leave one out" training and testing strategy. Besides being the most appropriate testing method for a dataset with only 15 individuals, it is also necessary for comparison with the other published works that use this dataset as any other training and testing methodology would produce incomparable results. This type of leave-one out cross validation is a special case of k-fold cross validation where the number of folds is the same as the number of distinct entities.

In this paper, error was determined as the average Euclidean distance between the ground truth and predicted angles on the left-out, person-exclusive test set as follows.

1) For persons 0 to n (where n is the number of distinct individuals in the dataset).
    a) Train a model on images and labels from all identities except for the one chosen above such that the model outputs predicted gaze directions (ϕ, θ).
    b) Test model using eye images from selected ("left-out") identity.
    c) For each prediction (ϕ, θ) and ground truth ($\hat{\phi}$, $\hat{\theta}$) above calculate the Euclidean distance between them.
    d) The mean value of the above distances is the error for this fold.
2) The sum of all the errors is divided by the number of folds to find the mean error – The error reported for a given neural network architecture is the mean error for all the folds.

Multiple deep neural networks were compared for eye gaze estimation. The publicly available MPII Gaze dataset was used for all experiments except for the first, where the UT Multiview dataset is also used.

A. Conventions for CNN Diagrams and Figures

A number of conventions for describing the layers of neural networks have arisen in deep learning literature. The authors of this paper choose to follow a common variant of these conventions. In general, the convention is that the type of layer is followed by the number of output parameters in parentheses. For example Conv(40) would indicate a convolutional layer with 40 output layers. When it is necessary to know the kernel size, this is also indicated by an @ symbol followed by the kernel size. For example, Conv(40@3×3) would indicate a convolutional layer with 40 outputs and a 3×3 kernel size.

When a layer type is followed immediately by a number, this number indicates the relative position of the layer compared to previous and subsequent layers of the same type. For example, conv1 indicates that this is the first convolutional layer, whereas conv2 would be the second convolutional layer.

The input layer is a special type of layer that describes how raw data is organized. In the case of this paper, the input layers take either eye crops or head poses. For input layers that take eye crops, the following format is used: Input([number of channels]×[image width]×[image height]).

Head pose inputs are not images but are instead simply angle values represented as floating point numbers. Input(ϕ, θ) means that the phi and theta values are being passed to the network at that point in the figure.

In the case of pooling operations the output size is determined by the input size divided by the size of the pool parameter. Max Pool(2×2) would indicate a 2×2 max pooling operation on the previous layer.

RELU is defined as max(0, ×) where × is the input. In all cases the size of the output of a relu layer is equal to the size of the output of the preceding layer.

Dropout layers are a regularization technique that helps prevent over-fitting and also increases the resiliency of a network. It is only used during training. Dropout randomly sets a percent of the weights in the network to zero. For example, dropout(0.5) indicates that 50% of the weights in the preceding layer should be set to zero in a random fashion. The output of dropout is always the same size and shape as the layer on which it operates.

FC or fc indicates a fully connected layer and fc(100) would indicate a fully connected layer with 100 outputs. In such layers, every "neuron" or unit is connected to every neuron of the previous layer. It is common to use such a layer after all the convolutional layers. For more detailed information about these layers, the reader is encouraged to consult the reference section, particularly [38], which contains a detailed description of the common types of layers used in modern deep learning approaches.

B. MPII Gaze Dataset Details

The MPII gaze dataset is a large collection of 213,659 images captured under unconstrained conditions from 15 subjects over several days. The images are collected under multiple illumination conditions. Some of the subjects wear spectacles and some do not. The images were captured at various gaze angles, recorded by software running on the participant's laptops. In each session, the subjects were asked to look at random sequences of 20 onscreen positions and to confirm their attentiveness, the subjects were asked to press the space bar once the onscreen target was disappearing.

The dataset contains eye and head features and target (gaze angle) values for every participant. To use MPII Gaze, the authors suggest mapping their reported vector to angles using a Rodrigues transformation, and this has been done for all reported experiments. The Rodrigues transformation is given by the Rodrigues rotation formula which provides a rotation matrix from which roll, pitch, and yaw can be generated [39].

IV. Experiments and Results

In this section, multiple experiments are described to provide insight on 4 primary research questions. These are tested on multiple CNN architectures and are discussed in this section. One of the first research goals was to achieve state of the art test error on a network that could perform inference within 3-15 ms on a typical single proprietary low power consumer embedded device.

The specific research questions are:

1) How does an architecture that uses both eyes compare to one that uses one eye in terms of accuracy?
2) How does simulated camera distance impact eye gaze accuracy for the proposed model?
3) Can augmentation be used to reduce any negative impacts?
4) Can the proposed hardware-friendly architecture perform with sufficient accuracy and speed?

TABLE I
RESULTS OF APPROACH 1

Training	Error
MPII Left eye only	5.9 degrees
MPII+UT Left eye only	5.6 degrees
MPII Both eyes	7.4 degrees
MPII+UT Both eyes	6.5 degrees
MPII Right eye only	5.3 degrees
MPII+UT Right eye only	6.1 degrees

Analysis of the errors due to choice of input during training (left, right, or flipped (both)). Training a network on both eyes using the flipping approach suggested in [27] appears to be a source of error. This observation informs the remaining experiments in the paper, which use a 2 channel approach instead.

First, the intra-dataset, person-exclusive experiments from Zhang *et al.* [27] were duplicated. The same procedure to estimate accuracy was used except the altered accuracy layers were modified to eliminate not a number (NaN) errors by replacing undefined values of the arc cosine function with the largest or smallest valid values, as appropriate.

A. Approach 1: Analysis of Eye Flipping

In Zhang *et al.* [27], one network is used for both eyes (with one inference per eye) and one of the eyes is flipped so that the gaze angle is roughly correct. An experiment was designed to see if this flipping had an impact on model accuracy. Six experiments were performed using the UT and MPII-Gaze datasets to see if training on both eyes or just one eye impacted accuracy. By doing experiments with combined and non combined datasets, it was also possible to determine weather they had similar distributions, and thus, if combining the two would be helpful for future experiments.

As shown in Table I, the method of individually classifying eye images and simply adjusting the right eye and angles as used in Zhang *et al.* [27] is a limiting factor in accuracy for that method. In both datasets, the performance was increased exclusively using left eyes or right eyes. This suggests that simply flipping the eye as suggested by Zhang *et al.* [27] may be a source of error in their model.

These results also indicated that the distribution of MPII Gaze and UT-Multiview are sufficiently different that combining the two for training gives no, or very little, improvement. Because of this, it was decided to use only MPII Gaze for the remaining experiments in this section as UT-Multiview had no significant influence on error.

B. A New Approach: Dual Eye Channels

Given the problems identified in the previous subsection with flipping one of the eyes, and not wanting to use two different networks for reasons of efficiency, a new approach involving using both the left and right eyes in separate input channels was investigated. Specifically, the left eye and right eye images are passed to the network in channels 0 and 1 respectively, and the gaze and pose information are averaged between the left and right eye images to create a single gaze and pose vector. Due to the results in the previous section, which indicated that data from UT did not significantly impact the results, only the MPII Gaze dataset was used.

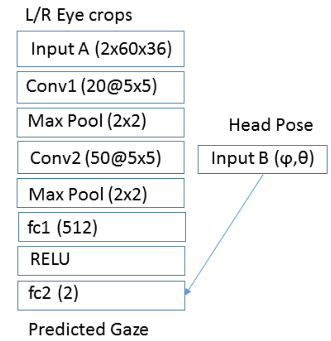

Fig. 1. Diagram of dual eye channel CNN used for eye gaze estimation. In contrast to previously published approaches, the right and left eye crops are input to the network simultaneously along with a single head pose vector. The output of the network is a single predicted pitch and yaw angle for both eyes, as if it were a single eye located exactly between the right and left eyes looking at the same object. This network was used for the experiment in Section IV-B and it was found to increase accuracy over previously published models. Please see Section III-A for a description of what the text in each of these layers means.

This modified, two channel architecture resulted in a significant increase in accuracy, averaging 4.63 degrees of error between the target and predicted values on unseen individuals. A diagram of this network can be seen in Fig. 1.

C. Efficiency: Can We Reduce the Number of Parameters?

Deep neural networks can often be made more efficient by reducing the number of parameters but this can sometimes come at the cost of accuracy. To see if reducing the number of parameters was possible without harming accuracy, an experiment was performed to halve the size of all output parameters. This experiment was not allowed to run for the full duration because the exact angle accuracy did not matter, only evidence that the network complexity could be reduced to a point where it would be small enough if necessary. This resulted in an average error of 4.980% on an unseen individual from the MPII Gaze dataset and indicates that reducing the number of parameters had little impact on accuracy.

D. Multi Resolution Experiments

Eye gaze systems in consumer devices must be able to maintain accuracy at a large range of distances. Although MPII Gaze has some variability in distance from the camera, the distances are not realistic for the conditions expected in, for example, a driver monitoring system or a distant cell phone camera. An experiment was conducted to determine if the network was able to perform under an expectedly wide range of subject distances.

Specifically, the goal was to accommodate realistic distances between the camera and the subject in situations that would be typical in commercial eye trackers that utilize low cost, low resolution cameras. To simulate the loss of information caused by distance, down-sampling using nearest neighbor interpolation was performed on the eye images in MPII Gaze as follows.

- Input image 60 × 36 ->Downscale to 52 × 31 ->Upscale to 60 × 36 ->CNN Eye gaze angle.

TABLE II
RESULT OF DISTANCE SIMULATION AND AUGMENTATION EXPERIMENTS

Resolution	Unaugmented error	Augmented error
60 x 36	4.63 degrees	4.918
52 x 31	9.90 degrees	4.94
26 x 16	10.10 degrees	4.98

This table shows the impact of camera distance and augmentation for a trained eye gaze model on angle accuracy using the network from experiment 2.

- Input image 60 × 36 ->Downscale to 26 × 16 ->Upscale to 60 × 36 ->CNN Eye gaze angle.

As can be seen in Table II, the network learned a narrow range of distances, and performance deteriorates when the subject is further from the camera than those in the training set. As a sanity check, an experiment was done to see if the downscaling algorithm was at fault for the poor results, so in addition to nearest, we also tried bicubic, linear, and LANCZOS using open source computer vision software. The experiment showed that the downscaling algorithm used had no influence on the results.

This demonstrates that the model is sensitive to changes in distance. In the next section, an experiment is performed to see if data augmentation can be used to improve upon this.

E. Impact of Random Resizing as Augmentation

Data augmentation has been shown in many studies [40] to have a large impact on model performance but augmenting to increase accuracy of on a wide range of distances appears to be neglected in literature on eye gaze. To further improve accuracy, the dataset was augmented with multiple randomly chosen resolutions to match the full range of desired distances. To help reduce the chance that the network would learn the specific interpolation method used, Nearest is used in the training set, but Lanczos filtering is used in the testing set.

These results indicate that augmenting the images with distances that are likely to be encountered in real world usage situations is an effective way to increase accuracy and succeeds in achieving some invariance to subject distance.

F. A Quest for Hardware Efficiency and Even Better Accuracy

It was shown by Simonyan and Zisserman [41] that two stacked layers of 3 × 3 convolutions have the same receptive field as a single 5×5 layer, with fewer multiplications. Because the efficiency of a convolutional neural network is primarily dependent on the number of multiplications, and thus the number of convolutions, stacked 3×3 convolutions were chosen for this experiment and the network was redesigned accordingly. Several experiments involving architectures with stacked 3×3 kernels were performed using different parameters. The best two architectures were further evaluated, and the best models from each of them were chosen and evaluated on multiple resolutions as shown in Table III. A diagram of the final architecture used can be seen in fig. 2, and a comparison with other published works can be seen in Table IV.

TABLE III
ERROR OF PROPOSED MODEL FOR VARIOUS RESOLUTIONS (DEGREES)

Model	36 x 60	31 x 52	26 x 16
Best	3.650	4.1690	4.240
Second best	4.100	4.324	4.366
Benchmark	4.917	4.940	4.970

The benchmark method is the best performing network from experiment 2, trained and tested with the same techniques. The second best model is called Network 4 in subsection H of this section and the best model is network 5.

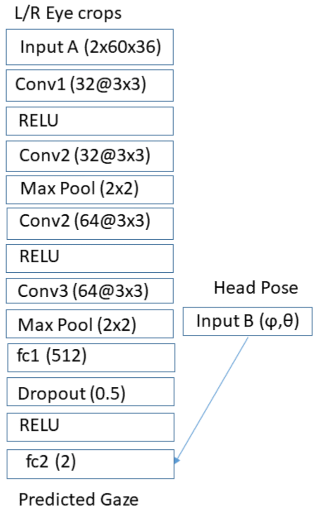

Fig. 2. This figure contains a diagram of the proposed network. It was found to further improve error over the method developed in Section IV-B from 4.63 degrees to 3.65 degrees. Like the network in fig. 1, this network takes both right and left eye image crops as input, using 2 channels and uses the same method to merge the left and right eye gaze vectors for the ground truth data. Another difference is the stacked 3 × 3 convolutions replacing the 5×5 convolutions and the addition of RELU nonlinearities The error of this network is compared with the error from other experiments in this paper in Table III. Please see Section III-A for a description of what the text in each of these layers means.

In order to maximize efficiency on a particular proprietary DSP, it was necessary to alter the output sizes of this network to be powers of 2. This is the explanation for why the output sizes are increased for this network compared to the others evaluated previously.

G. Runtime Analysis of the Used Networks

In this section, the execution time of the 5 networks used in this paper are evaluated on commodity processors. The result of these experiments are shown in Table IV.

Network 1 is the "one channel" approach used in [27] which we compare with. In contrast to the new approaches, this method performs a separate inference for each eye and flips the right eye.

Network 2 is the two channel improvement discussed in experiment 2 and shown in figure 1.

Network 3 is the variation of network 2 with half as many outputs per layer which was used in experiment 3.

Network 4 is a version of network 2 with 3×3 kernels but with the same number of units per layer as network 2. It is included to allow the reader to differentiate runtime that

TABLE IV
FRAMES PER SECOND OF CNN ON COMMODITY HARDWARE

Model	Embedded	CPU	GPU
Network 1 [27]	20.64	473.11	3984.09
Network 2	39.81	960.06	8078.47
Network 3	92.59	3080.80	12607.90
Network 4	29.68	592.02	6134.02
Network 5	19.11	400.50	5500.37
VGG16 [41]	0.0043	0.64	120.4

All units are in frames per second. Details of these networks are discussed in subsection H. As can be seen from this table, networks 2-4 perform faster on all tested hardware than the comparison networks (Network 1 and VGG16) while providing greater accuracy. Network 4 has similar performance to network 1 while increasing accuracy. The well known VGG16 network is included to allow the reader to see the speed gain over this type of architecture. VGG16 is an improvement over Alexnet, which was used to achieve 4.8 percent accuracy on the same dataset [25].

TABLE V
COMPARISON OF PROPOSED MODEL WITH OTHER PUBLISHED WORKS ON THE MPII-GAZE DATABASE

Citation	Error (degrees)
Baltrusaitis et al [36]	9.96
Wood et al [34]	9.58
Shrivastava et al [42]	7.8
Nie et al [33]	7.1
Zhang et al [27]	6.1
Zhang et al [25]	4.8
Proposed	3.65

This table shows the best performing network from this paper (Proposed) compared with other reported results on the MPII-Gaze dataset.

resulted from increasing the number of outputs from runtime that resulted from the change in network architecture.

Network 5 is the proposed model that has the best accuracy and utilizes the 3×3 kernels instead of 5×5 kernels. A diagram of this network is shown in figure 2.

The embedded processor is a typical 64 bit processor used in embedded and mobile systems, and available on a well-known embedded prototyping platform. Tests used only one core of this processor. The second processor is a popular high end workstation central processing unit (CPU). For a fair comparison, only one thread was used for tests. Lastly the GPU used is a popular high end consumer GPU targeted at video gamers. As can be seen in Table IV, network 3 is significantly faster than the other networks while network 5 provides a good balance between speed and accuracy. In some cases network 3 may be preferred due to increased speed and competitive accuracy.

H. Contributions

When deploying eye gaze solutions for consumer devices there are two important aspects to consider: Accuracy and efficiency. This paper contributes to addressing both issues by demonstrating improved accuracy and also by reducing the number of multiplications needed for predictions, thus increasing efficiency. It should be noted that the total number of matrix multiplications needed to obtain predictions from a convolutional neural network is determined by the size of the convolutional kernels, the step size, the number of nodes in each layer, and the number of layers [38]. These multiplications are often measured in multiply-accumulate operations (MACs).

For a digital signal processor (DSP) that primarily performs deep learning inference, the power consumption requirements are directly related to the number of MACs required per inference [43]–[45].

A variant of the proposed algorithm is now operating effectively on a hardware CNN SoC (system-on-chip) engine with 32 megabytes of random access memory, demonstrating that this approach is viable for low-resource consumer electronic devices.

At the beginning of this section we followed Zhang *et al.*'s approach [27] in Section IV-A. In Section IV-B, we found that by changing the network architecture to accept two eye crops, one for the left and one for the right eye in two input channels, and merging the gaze vectors and the position vectors, we were able to improve accuracy over that reported in Zhang *et al.* [27]. It was then shown in Section IV-C that the architecture introduced in Section IV-B could be made more efficient by halving the number of nodes in each layer with a negotiable decrease in accuracy.

Section IV-D demonstrated that the error more than doubles when the network is exposed to images that have been artificially resized to simulate a wide range of distances between the subject and the camera. This problem had not been addressed in previous work. It was then shown in Section IV-E that this increase in error could be nearly eliminated by use of data augmentation. Finally, the results of the experiment conducted in Section IV-F suggest a new network architecture that outperforms previously published works while reducing the number of multiplications and thus increasing efficiency.

V. CONCLUSION

Our results show that using information from both eyes in the neural network can increase accuracy. This is demonstrated in Section V, where adding additional eye information from the opposite eye enabled improved results over individual eyes, helping the network make sense of low quality images with ambiguous gaze. As expected, in all cases, the deeper network had the best performance. This research demonstrated the sensitivity of such models to variations in distance and how data augmentation can be used to overcome this. Most importantly, a new compact hardware-friendly architecture designed for use in small consumer electronics has been introduced and evaluated on the eye gaze task.

When evaluated on MPII Gaze, the proposed model performs favorably (see Tables IV and V) even when compared with much larger networks in the literature.

Running an optimized CNN based algorithm, as described in this paper can provide a high-performance, low-energy solution for continuous eye-tracking in next generation consumer electronic products such as ultra-lightweight smartphones and augmented reality glasses.

Since augmentation resulted in a significant improvement in accuracy, it may be fruitful to try other types of augmentation such as Generative Adversarial Networks (GANS) with landmarks, [46] and Smart Augmentation (SA) [40] in future work.

This will either require modifying such methods to work on regression problems, or translating the eye gaze problem into a classification task for the purpose of generating augmented data and then back to a regression task. Additionally, there are plans to investigate whether temporal information [47] can be used to further increase the accuracy without sacrificing the need for performance, as it has been shown to increase performance in DMS systems.

References

[1] Y. Liang, M. L. Reyes, and J. D. Lee, "Real-time detection of driver cognitive distraction using support vector machines," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 2, pp. 340–350, Jun. 2007.

[2] E. Wood and A. Bulling, "EyeTab: Model-based gaze estimation on unmodified tablet computers," in *Proc. Symp. Eye Tracking Res. Appl. (ETRA)*, 2014, pp. 207–210. [Online]. Available: http://doi.acm.org/10.1145/2578153.2578185

[3] A. Tawari and M. M. Trivedi, "Robust and continuous estimation of driver gaze zone by dynamic analysis of multiple face videos," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2014, pp. 344–349.

[4] V. Vaitukaitis and A. Bulling, "Eye gesture recognition on portable devices," in *Proc. ACM Conf. Ubiquitous Comput. (UbiComp)*, Pittsburgh, PA, USA, 2012, pp. 711–714. [Online]. Available: http://doi.acm.org/10.1145/2370216.2370370

[5] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[6] J. Lemley, S. Bazrafkan, and P. Corcoran, "Deep learning for consumer devices and services: Pushing the limits for machine learning, artificial intelligence, and computer vision," *IEEE Consum. Electron. Mag.*, vol. 6, no. 2, pp. 48–56, Apr. 2017.

[7] N. P. Jouppi *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Proc. ACM/IEEE 44th Annu. Int. Symp. Comput. Archit. (ISCA)*, 2017, pp. 1–12.

[8] D. W. Hansen and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 3, pp. 478–500, Mar. 2010.

[9] A. Kar and P. Corcoran, "A review and analysis of eye-gaze estimation systems, algorithms and performance evaluation methods in consumer platforms," *IEEE Access*, vol. 5, pp. 16495–16519, 2017.

[10] F. L. Coutinho and C. H. Morimoto, "Augmenting the robustness of cross-ratio gaze tracking methods to head movement," in *Proc. Symp. Eye Tracking Res. Appl.*, 2012, pp. 59–66.

[11] S. Chen and C. Liu, "Eye detection using discriminatory haar features and a new efficient SVM," *Image Vis. Comput.*, vol. 33, pp. 68–77, Jan. 2015.

[12] H.-C. Lu, C. Wang, and Y.-W. Chen, "Gaze tracking by binocular vision and LBP features," in *Proc. 19th Int. Conf. Pattern Recognit.*, Tampa, FL, USA, Dec. 2008, pp. 1–4.

[13] Z. Zhu and Q. Ji, "Eye and gaze tracking for interactive graphic display," *Mach. Vis. Appl.*, vol. 15, no. 3, pp. 139–148, Jul. 2004. [Online]. Available: https://doi.org/10.1007/s00138-004-0139-4

[14] C. Jian-Nan, Z. Chuang, Y. Yan-Tao, L. Yang, and Z. Han, "Eye gaze calculation based on nonlinear polynomial and generalized regression neural network," in *Proc. 5th Int. Conf. Nat. Comput.*, vol. 3, Aug. 2009, pp. 617–623.

[15] J. Wang, G. Zhang, and J. Shi, "2D gaze estimation based on pupil-glint vector using an artificial neural network," *Appl. Sci.*, vol. 6, no. 6, p. 174, 2016. [Online]. Available: http://www.mdpi.com/2076-3417/6/6/174

[16] Z. Zhu, Q. Ji, and K. P. Bennett, "Nonlinear eye gaze mapping function estimation via support vector regression," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, vol. 1. Hong Kong, 2006, pp. 1132–1135.

[17] J. Zhu and J. Yang, "Subpixel eye gaze tracking," in *Proc. 5th IEEE Int. Conf. Autom. Face Gesture Recognit.*, Washington, DC, USA, May 2002, pp. 124–129.

[18] E. D. Guestrin and M. Eizenman, "General theory of remote gaze estimation using the pupil center and corneal reflections," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 6, pp. 1124–1133, Jun. 2006.

[19] T. Ohno and N. Mukawa, "A free-head, simple calibration, gaze tracking system that enables gaze-based interaction," in *Proc. Symp. Eye Tracking Res. Appl. (ETRA)*, San Antonio, TX, USA, 2004, pp. 115–122. [Online]. Available: http://doi.acm.org/10.1145/968363.968387

[20] Z. Zhu and Q. Ji, "Novel eye gaze tracking techniques under natural head movement," *IEEE Trans. Biomed. Eng.*, vol. 54, no. 12, pp. 2246–2260, Dec. 2007.

[21] D. Beymer and M. Flickner, "Eye gaze tracking using an active stereo head," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2003, p. 451.

[22] J. Lemley, A. Kar, and P. Corcoran, "Eye tracking in augmented spaces: A deep learning approach," in *Proc. IEEE Games Entertainment Media Conf. (GEM)*, Galway, Ireland, 2018, pp. 385–390.

[23] Y. Fu, W.-P. Zhu, and D. Massicotte, "A gaze tracking scheme with low resolution image," in *Proc. IEEE 11th Int. New Circuits Syst. Conf. (NEWCAS)*, Paris, France, Jun. 2013, pp. 1–4.

[24] A. George and A. Routray, "Real-time eye gaze direction classification using convolutional neural network," in *Proc. Int. Conf. Signal Process. Commun. (SPCOM)*, Jun. 2016, pp. 1–5.

[25] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "It's written all over your face: Full-face appearance-based gaze estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Honolulu, HI, USA, Jul. 2017, pp. 2299–2308.

[26] H. Deng and W. Zhu, "Monocular free-head 3D gaze tracking with deep learning and geometry constraints," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3162–3171.

[27] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 4511–4520.

[28] K. Krafka *et al.*, "Eye tracking for everyone," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2176–2184.

[29] H. Park and D. Kim, "Gaze classification on a mobile device by using deep belief networks," in *Proc. 3rd IAPR Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2015, pp. 685–689.

[30] C. Zhang, R. Yao, and J. Cai, "Efficient eye typing with 9-direction gaze estimation," *Multimedia Tools Appl.*, vol. 77, no. 15, pp. 19679–19696, Nov. 2017. [Online]. Available: https://doi.org/10.1007/s11042-017-5426-y

[31] X. Wu, J. Li, Q. Wu, and J. Sun, "Appearance-based gaze block estimation via CNN classification," in *Proc. IEEE 19th Int. Workshop Multimedia Signal Process. (MMSP)*, Oct. 2017, pp. 1–5.

[32] S. D. Iyer and H. Ramasangu, "Hybrid LASSO and neural network estimator for gaze estimation," in *Proc. IEEE Region 10 Conf. (TENCON)*, Singapore, Nov. 2016, pp. 2579–2582.

[33] S. Nie, M. Zheng, and Q. Ji, "The deep regression Bayesian network and its applications: Probabilistic deep learning for computer vision," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 101–111, Jan. 2018.

[34] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling, "Learning an appearance-based gaze estimator from one million synthesised images," in *Proc. 9th Biennial ACM Symp. Eye Tracking Res. Appl. (ETRA)*, Charleston, SC, USA, 2016, pp. 131–138. [Online]. Available: http://doi.acm.org/10.1145/2857491.2857492

[35] E. Wood *et al.*, "Rendering of eyes for eye-shape registration and gaze estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 3756–3764. doi: 10.1109/ICCV.2015.428.

[36] T. Baltrusaitis, P. Robinson, and L.-P. Morency, "OpenFace: An open source facial behavior analysis toolkit," in *Proc. IEEE Win. Conf. Appl. Comput. Vis. (WACV)*, Mar. 2016, pp. 1–10.

[37] U. Weidenbacher, G. Layher, P.-M. Strauss, and H. Neumann, "A comprehensive head pose and gaze database," in *Proc. 3rd IET Int. Conf. Intell. Environ.*, Sep. 2007, pp. 455–458.

[38] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1. Cambridge, MA, USA: MIT Press, 2016.

[39] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*, vol. 201. Englewood Cliffs, NJ, USA: Prentice-Hall, 1998.

[40] J. Lemley, S. Bazrafkan, and P. Corcoran, "Smart augmentation learning an optimal data augmentation strategy," *IEEE Access*, vol. 5, pp. 5858–5869, 2017. [Online]. Available: https://doi.org/10.1109/ACCESS.2017.2696121

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556 [cs.CV]*, Sep. 2014.

[42] A. Shrivastava *et al.*, "Learning from simulated and unsupervised images through adversarial training," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 2242–2251.

[43] C. Turner, "Calculation of TMS320LC54x power dissipation," Texas Instrum., Dallas, TX, USA, Rep. SPRA164, 1997.

[44] J. Lemley, S. Bazrafkan, and P. Corcoran, "Learning data augmentation for consumer devices and services," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Las Vegas, NV, USA, 2018, pp. 1–3.

[45] A. Abnous, K. Seno, Y. Ichikawa, M. Wan, and J. Rabaey, "Evaluation of a low-power reconfigurable DSP architecture," in *Proc. Int. Parallel Process. Symp.*, 1998, pp. 55–60.

[46] S. Bazrafkan, H. Javidnia, and P. Corcoran, "Face synthesis with landmark points from generative adversarial networks and inverse latent space mapping," *arXiv:1802.00390 [eess.IV]*, Feb. 2018.

[47] J. Lemley, S. Bazrafkan, and P. Corcoran, "Transfer learning of temporal information for driver action classification," in *Proc. 28th Mod. Artif. Intell. Cogn. Sci. Conf. (MAICS)*, 2017, pp. 123–128.

Joseph Lemley (S'16) received the B.S. degree in computer science and the M.S. degree in computational science from Central Washington University, Ellensburg, WA, USA, in 2006 and 2016, respectively. He is currently pursuing the Ph.D. degree in electronic engineering with the National University of Ireland, Galway.

He is a Research and Development Engineer with Fotonation, Galway, Ireland, under the IRCSET Employment Ph.D. Program. His research interests include artificial intelligence, deep learning, and computer vision.

Mr. Lemley was a recipient of the 2017 Best Paper Joint Award for *IEEE Consumer Electronics Magazine*, the Best Paper Second Place Award at ICCE 2018, and other awards during previous years.

Anuradha Kar (S'08) was born in Kolkata, India. She received the Bachelor of Technology degree in electronics and communication engineering from the West Bengal University of Technology, Kolkata, in 2009 and the Master of Technology degree in radio, physics and electronics from the University of Calcutta, Kolkata, in 2011. She is currently pursuing the Ph.D. degree with the National University of Ireland, Galway.

Her research interests include human–computer interaction, application of eye gaze in next generation consumer applications like AR/VR and smart devices, and performance evaluation of sensor systems.

Ms. Kar was a recipient of the Second Position at the All India IEEE Student Project Contest in 2009 and two year Master Degree Fellowship from the Indian Space Research Organization from 2009 to 2011.

Alexandru Drimbarean (M'16) received the B.S. degree in electronic engineering from the Transilvania University of Brasov, Romania, in 1997 and the M.Sc. degree in electronic engineering from the National University of Ireland, Galway, Ireland, in 2002.

He has been the Vice President of Advanced Research with Fotonation, Galway, since 2015 and has worked with Fotonation in various research, engineering, and management roles since 2000. He has authored several journal articles as well as over 30 patents. His interests include image processing, and understanding as well as computer vision and machine learning.

Peter Corcoran (M'95–F'10) received the B.A.I. degree in electronic engineering, the B.A. degree in mathematics, and the Ph.D. degree in electronic engineering from Trinity College Dublin, Ireland, in 1984 and 1987, respectively, focusing on dielectric liquids.

He is a Professor with the Department of Electrical and Electronic Engineering, National University of Ireland, Galway, where he has taught since being appointed to a lectureship in 1986. He has co-authored over 350 technical publications and has co-invented over 300 granted U.S. patents. His research interests include biometrics, imaging, deep learning, and edge-AI and consumer electronics.

Prof. Corcoran was a recipient of the numerous industry and academic awards and honors. He is the Past Editor-in-Chief of *IEEE Consumer Electronics Magazine*.