A Comparative Study of DeepLabCut and Other Open-Source Pupillometry Data Analysis Algorithms – Which to Choose?

Amitesh Badkul[1,3], Sonakshi Mishra[1], and Srinivasa P. Kommajosyula[2,*]

[1] *Department of Electrical & Electronics Engineering, Birla Institute of Technology and Science, Hyderabad Campus, Jawahar Nagar, Hyderabad, India*

[2] *Department of Pharmacy, Birla Institute of Technology and Science, Hyderabad Campus, Jawahar Nagar, Hyderabad, India*

[3] *Ph.D. Program in Computer Science, The Graduate Center, The City University of New York, New York, NY, United States of America*

[*] *Corresponding author: Srinivasa P. Kommajosyula (ksprasad@hyderabad.bits-pilani.ac.in)*

Abstract Pupillometry measures pupil size, and several open-source algorithms are available to analyse pupillometry data. However, only a few studies compared these algorithms' accuracy and computational resources. This study aims to compare the accuracy of computer vision-based algorithms (Swirski, Starburst, PuRe, ElSe, ExCuSe algorithms) and the machine learning algorithm, DeepLabCut, to the double-blinded human examiners (gold-standard). Training of DeepLabCut with different architectures and a variable number of markers (2-9 markers) was done on an open-source dataset. The duration of training was statistically longer for the ResNet152 model compared to the MobileNet model. The pupil diameters in computer vision-based software such as PuRe, Starburst, and Swirski were statistically different from human measurements. MobileNet 2 and 3 marker models were the closest to the human measurements. In conclusion, this work highlights the efficiency of lower marker models based on MobileNet architecture in DeepLabCut, which consumes fewer computational resources and is more accurate.

Keywords: machine learning, deep learning, pupillometry, DeepLabCut, MobileNet, computer vision.

1. Introduction

Pupillometry measures pupil size changes in response to external stimuli or internal states [8, 33]. Pupil size changes with bright light, cognitive load, attention, memory, internal state, emotional and neuromodulatory changes [15,16,19]. Pupillometry is used both clinically and in basic science research to evaluate neurological function and in the diagnosis of attentional disorders [10, 30]. Neuroscience research in both animal models and humans has identified that an increase in activity at *locus coeruleus* and release of norepinephrine are causal in pupil diameter changes. Some researchers even use pupil diameter as a surrogate for *locus coeruleus* activity [6, 20].

Most methods of pupillometry use conventional image processing-based techniques like segmentation, edge detection, and ellipse fitting with thresholding followed by contour fitting. Recent studies, however, have employed sophisticated machine learning-based algorithms such as convolutional neural networks or generative adversarial networks, as these machine learning algorithms give superior accuracy values. [2,4,14,21,28].

Some studies have analysed the efficacy of these analytical techniques to determine the computational demand as well [21]. Many open-source pupillometry software are available that use various mechanisms to analyse pupil diameter, such as PupilEXT, Starburst, ExCuSe, ElSe, PuRe, and PuReST [7,35]. A recent study validated this software and found that the ExCuSe, ElSe, PuReST, and PuRe algorithms attained adequate accuracy for pupil diameter measurement [35]. The Starburst algorithm detected many false peaks and produced highly variable results. The Swirski algorithm failed to detect the pupil in the 630nm spectrum.

Among machine learning applications, both supervised and unsupervised learning approaches are present [13]. A recent experiment using a deep learning algorithm called DeepLabCut garnered interest due to its applicability in a variety of experimental variables, broad user base with active software development, and ease of use due to its graphical user interface based on Python [3]. Here, the experimenter manually places the markers on the pupil diameter. It employs a transfer learning approach and requires less data (30 frames) compared to other approaches requiring thousands of frames for training, making it an attractive option for pupillometry data analyses [24]. DeepLabCut was initially developed as a pose estimation software in biology/behavioral/neuroscience research. It has also been used to detect animal behaviour and movement [18]. It has been applied to pupillometry research only recently [24]. Privitera et al. developed a low-cost (approximately 300 euros) Raspberry Pi setup and pupillometry software to analyse pupillometry data. They trained a machine learning model (Deep LabCut with 11 markers) using ResNet to quantify pupil diameter utilizing the DeepLabCut library. However, the reliability of DeepLabCut compared to other open-source software, the impact of the number of markers on the accuracy of the measurements, and the usage of computational resources are not known. Hence, this study is designed to address the following issues.

A. To assess the computational efficiency of DeepLabCut architectures and models.

B. To evaluate the accuracy of the DeepLabCut models (ResNet, and MobileNet architecture-based models with various markers ranging from 2-9 markers) in comparison to other open-source pupillometry software and human examiners measuring the pupil (considered as the gold-standard).

C. To benchmark current open-source algorithms against the gold-standard.

2. Methodology

2.1. Experimental design

We have used data from an open-source dataset published in a previous publication by Privitera et al. [24]. In brief, mice on C57 background ($n = 17$) of age 2-4 months were head restrained, and changes in pupil diameter were recorded using a Raspberry Pi

NoIR V2 camera under dark conditions with IR and UV lights. In the current manuscript, the DeepLabCut models were trained using 30 data frames and tested them on 20 different frames snipped randomly from different videos made by Privitera et al. Random test frames were selected to avoid temporal bias and capture different phases of pupil dilations. However, the same test frames were used to test all the models evaluated in this study. In a previous study employing DeepLabCut, only 30 frames or 150 frames of data were used [22, 24]. During the training phase, the computational resources being consumed were measured using the weights and biases tool (WandB [32]). After training DeepLabCut models based on various deep convolutional neural architectures like ResNet 50, ResNet 152, and MobileNetV2. Later, compared the pupil measurements of these DeepLabCut models to open-source algorithms as well as human examiner measurements of pupil diameter for accuracy check. The distance between two points on the pupil was measured to calculate the diameter, followed by inference of radii, and an average was taken in cases where multiple markers were used. Human examiners measured the pupil diameter in the same frames used for testing DeepLabCut and other open-source software using ImageJ version 1.53 (a National Institute of Health algorithm). The distance was measured in pixels and converted to millimeters using the ground-truth values derived by Privitera et al.

For measurement purposes, the data were measured in pixels and converted to mm. The distance between the two tracked calibration points in pixels was calculated using the formula:

$$d^{\,\mathrm{px}} = \sqrt{\left(x_{P_{c1}}^{\mathrm{px}} - x_{P_{c2}}^{\mathrm{px}}\right)^2 + \left(y_{P_{c1}}^{\mathrm{px}} - y_{P_{c2}}^{\mathrm{px}}\right)^2}\,, \tag{1}$$

where $x_{P_{ci}}^{\mathrm{px}}$, $y_{P_{ci}}^{\mathrm{px}}$, $i = 1, 2$, are x and y coordinates of the calibration points P_{c1}, P_{c2}, respectively, in pixels. Privitera et al. advise using median values for these calculations. The absolute dimension of the calibration object in mm ($d^{\,\mathrm{mm}}$) has to be divided by $d^{\,\mathrm{px}}$ to obtain the pixel-to-mm conversion ratio:

$$\mathrm{ratio}^{\,\mathrm{mm/px}} = \frac{d^{\,\mathrm{mm}}}{d^{\,\mathrm{px}}}\,, \tag{2}$$

and x and y coordinates of all tracked points P_j at each frame are multiplied by the conversion ratio, resulting in a metric description of the tracked points:

$$\begin{aligned} x_{P_j}^{\mathrm{mm}} &= x_{P_j}^{\mathrm{px}} \times \mathrm{ratio}^{\,\mathrm{mm/px}}\,, \\ y_{P_j}^{\mathrm{mm}} &= y_{P_j}^{\mathrm{px}} \times \mathrm{ratio}^{\,\mathrm{mm/px}}\,. \end{aligned} \tag{3}$$

All the data measured in pixels by human examiners, as well as the algorithms, were individually converted to millimeters using the above formulae.

2.2. DeepLabCut and deep convolutional neural networks (DCNNs)

DeepLabCut, a Python-based framework, was used to create the DCNN models. A variable number of markers from 2 to 9 were used to generate the training dataset for the DCNN models. The principle of this method is based on the concept of transfer learning, that is, training pre-trained models for a different task. The following models were used here: ResNet50, ResNet152, and MobileNet V2 [9, 12, 26]. After training, the DCNN models render the markers and output the location of each marker on each frame, from which the pupil radius is calculated. The hyperparameters used are mentioned below.

Markers Markers were used on the pupil to train the DCNN models, ranging from 2 markers to 9 markers. An additional two markers on the ends of the eyes were used to establish the ground truth. The reason for the usage of different types of labeling is to improve the accuracy while calculating the pupil radii.

Iterations The number of training iterations is 30,000.

Learning rate scheduler The learning rate used in training approach follows a multi-step schedule. The rate is adjusted at specific iterations, a strategy that enhances the training process's efficacy and performance.The learning rates were 0.005 and 0.020, while the respective iterations were 10000 and 30000.

Batch size The number of samples the model processes before each update, known as the batch size, is set to 1. Learning rate decay gradually diminishes the learning rate, preventing the model from overshooting the loss function's minima. In this case, for every 30,000 iterations, the decay is utilized.

Loss Lastly, the loss function used in this model is the Huber loss function, which integrates the properties of mean squared error loss and mean absolute error loss.

2.3. Other open-source software

Most prominent open-source softwares, such as Swirski, Starburst, PuRe, ElSe, and ExCuSe were used to compare their accuracy to that of the DeepLabCut models. These algorithms use template matching, edge detection, thresholding, or best-fit approaches.

Swirski This algorithm detects the light reflection and uses template matching followed by thresholding and edge detection techniques to estimate the size of the pupil. Readers are directed to the manuscript by Zandi et al. for a more detailed review [35]. The parameters used in this study for Swirski are mentioned here: Minimum radius: 20; Maximum radius: 140; Canny blur: 1.6; Canny threshold 1: 15; Canny threshold 2: 45; Perc inliners: 20; Inliner iterations: 2; Image Aware RANSAC: Yes.

Starburst This algorithm first detects the edges using Canny edge detection, followed by interpolation of the center from the detected edges. Readers are directed to the manuscript by Zandi et al. for a more detailed review [35]. The parameters used in this study for Starburst are mentioned here: Edge threshold: 21; Number of rays:

32; Minimum feature candidates: 7; CR Ratio (to image height): 10; CR window (px): 433.

PuRe This algorithm fits a model to detect pupil diameter. Readers are directed to the manuscript by Zandi et al. for a more detailed review [35]. The parameters used in this study for PuRe are mentioned here: Image width (downscaling): 320; Image height (downscaling): 240; Mean Canthi distance: 27.6; Maximum pupil size: 8; Minimum pupil size: 2; Minimum radius: 50.

ElSe This algorithm uses edge detection and thresholding techniques followed by ellipse fitting to identify the pupil diameter. Readers are directed to the manuscript by Zandi et al. for a more detailed review [35]. The parameters used in this study for ElSe are mentioned here: Minimum area (%): 0.005; Maximum area(%): 0.2.

ExCuSe The algorithm uses edge detection followed by mathematical estimations to calculate the pupil diameter. Readers are directed to the manuscript by Zandi et al. for a more detailed review [35]. The parameters used in this study for ExCuSe are mentioned here: Ellipse Goodness threshold: 15; Maximum radius: 50.

Non-deep learning algorithms are based on mathematical approaches and don't need any training. Testing time has been reported to be under 100 milliseconds per frame previously, and is similar to our anecdotal observations. Since comparing the accuracy of the test was the prime objective, testing times were not compared and this is a caveat that could be addressed in the future [27, 35]. In the current study, a standard PC was used (RAM: 8 GB DDR4-3200 MHz; CPU type AMD Ryzen™ 5 5600H.Graphics card: NVIDIA GeForce RTX 3060, Laptop GPU with 6 GB GDDR6 VRAM, storage: SSD/HD of 512 GB M.2 NVMe SSD).

ImageJ Examiners used ImageJ to measure the pupil diameter. Using line tool, and functions: analyse and measure, the pupil diameters were measured.

WandB WandB has been a powerful tool in logging the hyper-parameters involved in training a model over a dataset and storing the visualizations of a training run, analysing the different metrics that come into play while developing a model through training mainly when a graphics processing unit (GPU) is involved. This gives the developer an overall and comprehensive view of the process in play during training and also saves the training behaviour of GPU: power usage, memory allocated, time spent accessing memory, temperature and utilization for future references and comparisons. Training of DeepLabCut models was performed in Google Collab and recorded the GPU parameters using WandB as in the literature [1].

Statistical analyses A comparison of the accuracy of the detection of pupil diameter by various algorithms and human experimenters was done. A repeated measures ANOVA test followed by post-hoc Bonferroni correction for comparisons between the groups was used. T-tests and ANOVA were used to compare the computational resource usage. Data are represented with mean and standard error of the mean (SEM) values across this manuscript.

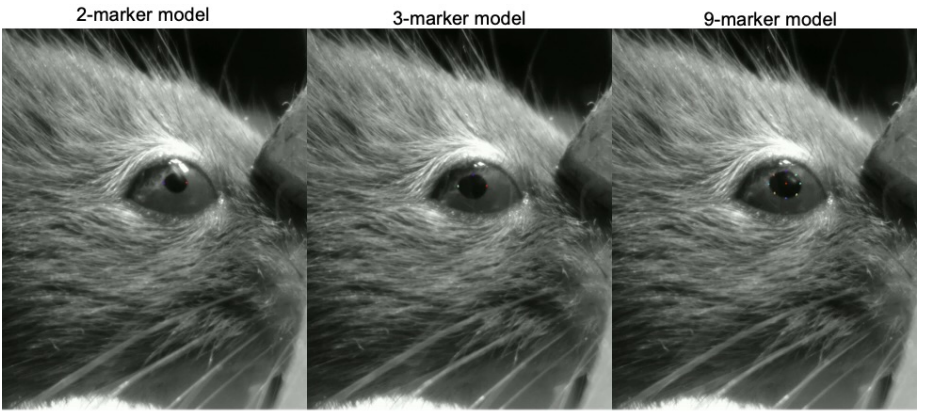


Fig. 1. Mouse pupils with overlaid markers for different architectures in an anesthetized and head-fixed mouse. (**a**) Pupil with two markers at both ends of the eye. (**b**) Pupil with three markers. (**c**) Pupil with nine markers placed around the eye.

3. Results

The training-related use of computational resources by these DCNN models in the DeepLabCut module was assessed. Firstly, one of the experimenters marked specific regions in the image as the borders of the pupils using a variable number of markers (Fig. 1).

The GPU use times, as a measure of computational resource usage, for ResNet50, ResNet152, and MobileNetV2 models were measured during training in DeepLabCut using WandB. In most cases, the GPU usage was within the range 80-95%; hence, this parameter was not statistically validated. There is a significant difference in GPU usage durations during training between all the network architectures (F-statistic = 846; $p < 0.001$). The post-hoc Bonferroni test showcased significant differences between all three network architectures in the usage of GPU resources during training. The GPU power usage duration was longer in ResNet152 models (range = $\langle 147.5, 153.5\rangle$ min, mean = 151.3 min) vs. ResNet50 models (range = $\langle 61.5, 78\rangle$ min, mean = 74.4 min). In comparison, MobileNet models were using the GPU resources for the least amount of time (range = $\langle 47, 70.5\rangle$ min, mean = 59.7 min) (Fig. 2).

These results show that MobileNet models, regardless of the number of markers, consume GPU resources for lesser duration compared to other neural network architectures. An individual analysis of these different network architectures across different marker models showed similar GPU consumption in ResNet152 (147.5, 151, 150, 151.5, 152.5, 151, 153.5 and 153.5 min) and ResNet50 (61.5, 74.5, 76, 77, 75.5, 77, 78 and 75.5 min)

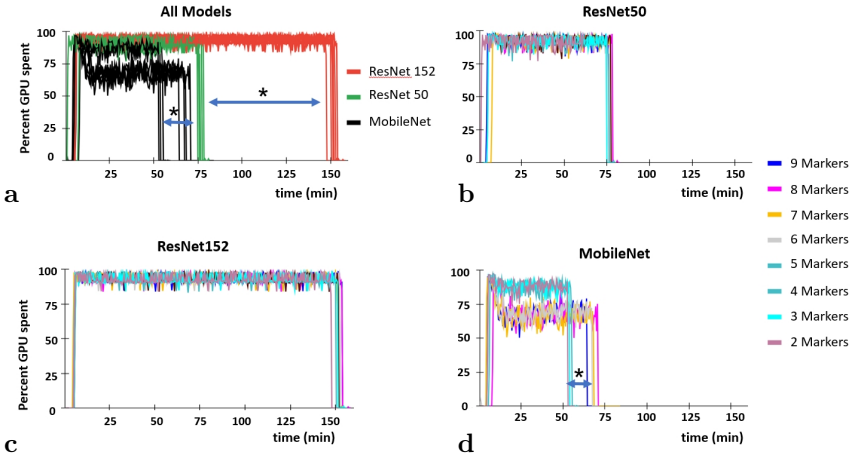


Fig. 2. Comparison of ResNet152, ResNet50 and MobileNet on the duration of use of graphics processing unit (GPU) during training in the DeepLabCut. (**a**) Three models together. MobileNet models utilize the least time, whereas the ResNet152 models the most and ResNet50 models are in between. (**b**) ResNet50 models showcase a similarity in GPU usage time across all models (2–9 markers). (**c**) ResNet152 models showcase a similarity in GPU usage time across all models (2–9 markers). (**d**) MobileNet models showcase a significant difference in GPU usage time for lower marker models (2–5 markers) vs. higher marker models (6–9 markers). Blue arrows with stars indicate the differences.

for different marker models (2-9 markers) (Fig. 2b and c). However, a decrease in GPU usage duration was noted for lower marker models (2, 3, 4 and 5: 52.5, 53.5, 47 and 55 min) vs. higher marker models (6, 7, 8, and 9: 67, 68, 70.5 and 64 min) in MobileNet architecture (Fig. 2d). The differences in GPU usage times were significantly different in MobileNet lower and higher models (Mean±SEM values, lower vs. higher: 52±1.74 min vs. 67.38±1.34 min, t-statistic $= 5.12, d_f = 3, p < 0.05$; d_f – number of degrees of freedom). This suggests that lower marker models of MobileNet architecture consume the least computational resources of all the DeepLabCut models.

After training the models, the accuracy of these models was tested on a set of 20 images from different mice (Fig. 3). As the gold standard for comparisons the human examiners' measurements of pupils were considered. The two examiners were double-blinded and weren't involved in any part of the experiments. Examiners used ImageJ software, and the values determined by the examiners were averaged to arrive at a single value and to enable statistical comparisons. The inter-examiner variability in measurements was less than 0.7% across all the frames. The mean value of pupil diameter found by the human examiners was 1.01 mm.

In addition to DeepLabCut models, the PuRe, Starburst, Swirski, ElSe, and ExCuSe

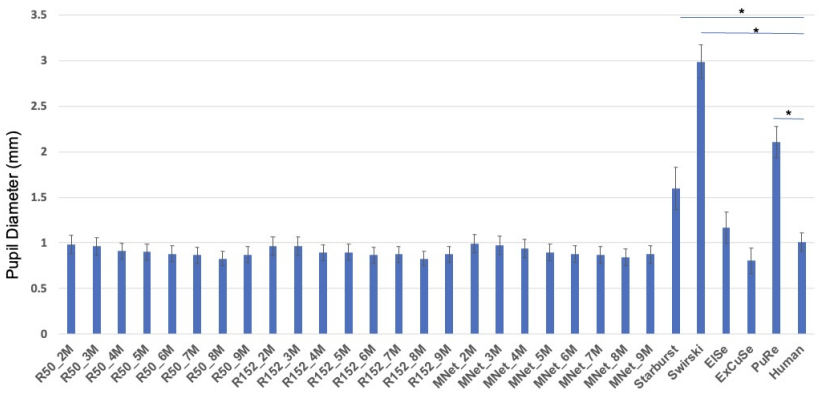


Fig. 3. Comparison of all DeepLabCut architecture models and computer vision-based models to the human examiner measures of pupil diameter. Three DeepLabCut-based architectures, including ResNet50, ResNet152, and MobileNet, were compared to computer vision-based models and human measurements. This includes: ResNet50 for 2–9 markers (R50_2M to R50_9M), ResNet152 for 2–9 markers (R152_2M to R152_9M), MobileNet for 2–9 markers (MNet_2M to MNet_9M), and human examiners (Human). Blue arrows with stars indicate significant differences between groups MobileNet vs. ResNet50; ResNet 50 vs. ResNet152 and lower marker models (2-5) vs. higher marker models (6-9) of MobileNet.

were also tested to compare these models' performance. The results of these models were significantly different from the human measurements of the pupil diameter ($p < 0.001$, Bonferroni post-hoc test, repeated measures ANOVA).

The repeated measures ANOVA showcased a significant difference between all the models (F-value $= 34.857, p < 0.001$). Post-hoc tests using Bonferroni corrections were performed for individual comparisons. All the DCNN-trained models were non-significantly different from the pupil diameter values measured by examiners. However, there was a significant difference in the pupil diameter measured using open-source algorithms such as PuRE, Starburst, and Swirski in comparison to examiners (Mean±SEM values: 2.104±0.17, 1.599±0.232, 2.987±0.185 vs. 1.01±0.1, $p < 0.001$, repeated measures ANOVA followed by post-hoc Bonferroni test). ExCuSe and ElSe were not significantly different in comparison to human examiners (Mean±SEM values: 0.805±0.141, 1.165±0.169 vs. 1.01±0.1). However, the mean values of ExCuSe and ElSe compared to the mean of examiners vary by 20.3% and 15.3%. All the machine learning models were non-significantly different from the examiner's measures. Among all the machine learning models, two marker models of ResNet50 (0.983±0.099), ResNet152 (0.964±0.098), as well as the MobileNet model (0.99±0.102), showcased the least variance from the examiners' mean pupil diameter (1.01±0.1) and their means varied by 2.6%, 4.5%, and 1.98%, respectively, from the mean measurement of examiners. Three marker models of

all architectures followed the two marker models closely (Mean±SEM values: ResNet152 – 3 marker: 0.966±0.099; ResNet50 – 3 marker: 0.961±0.097; MobileNet – 3 marker: 0.973±0.104). The 2 and 3 marker models were more accurate in detecting the pupil diameter than models with a higher number of markers.

The measurements of memory consumption made with WandB indicated that the DeepLabCut models, MobileNet models, specifically the lower marker models from 2 to 5 markers, consume less memory resources than other models.

4. Discussion

The data showcased that among the DeepLabCut models, MobileNet models, specifically the lower marker models from 2 to 5 markers, consume less memory resources. Mainly, the accuracy of these models were compared to open-source pupil measurement software and human observers. The accuracy of the MobileNet 2 marker model is shown to be closest to that of human observers. All open-source pupillometry software tested here has either overvalued or undervalued the pupil diameter compared to human experimenters. ElSe and ExCuSe were the closest in terms of performance as compared to human observers. DeepLabCut toolbox has been used mostly in animal pose estimation, and only recently has it been used to analyse pupil data [18, 24]. Results show that the error in pupil diameter increases with the increase in the number of markers across all three network architectures in the DeepLabCut. This error could be due to the intrinsic nature of deep learning models, where a balance between the number of markers (a surrogate for the learnable parameters), the use of definite architectures (a surrogate for the complexity of the models), and the number of frames used in training determine the learning efficiency and prevent over/underfitting. Since the amount of training data was the same in experiments for any architecture and number of markers (2 – 9), this could have impacted the model's ability to learn and led to over/underfitting. These may cause a decrease in the performance of models using a higher number of markers [11]. There is a difference in the amounts of parameters used by ResNet 50 (23.5 million) vs. ResNet 152 (58.3 million) vs. MobileNetV2 (3.4 million). The lower number of parameters in MobileNetV2 decreases the training duration/computational resources compared to other deep learning models. The depthwise convolutions with fewer parameters in MobileNetV2 increase efficiency and decrease computational costs [17, 26]. These findings suggest that while larger models (e.g., ResNet-152) offer a theoretical advantage in complex tasks due to their higher capacity and can handle vanishing gradient issues. While, the simpler MobileNetV2 architecture performed equally well for this specific task with much lower computational demand. This makes MobileNetV2 the most efficient choice for real-time pupillometry or scenarios where hardware resources are limited without a significant sacrifice in accuracy [29].

It is difficult to compare the mathematical superiority of DeepLabCut with other

techniques because each technique uses different mathematical approaches and principles. However, here is a comparison of some key features of DeepLabCut that make it stand out from the other techniques.

Deep learning Learning complex patterns and features from input data is common in DeepLabCut and other deep learning techniques. These traits increase the adaptability of DeepLabCut to different lighting conditions, pupil sizes, and head positions. Thus DeepLabCut is more robust and accurate than traditional computer vision algorithms.

Flexibility DeepLabCut provides flexibility and automatically extracts features that are unlike traditional computer vision algorithms, which often have fixed and predefined features.

Small training data DeepLabCut requires only a small amount of data for training as opposed to other machine learning algorithms, and it can also perform with better accuracy and robustness. This is because DeepLabCut can learn from a diverse set of examples and generalize to new conditions.

Real-time processing DeepLabCut can process images in real-time on a variety of platforms, making it suitable for applications that require accurate pupil detection [31].

Model architecture DeepLabCut models can be designed and optimized for specific tasks and data types. This allows for better performance and generalization compared to traditional computer vision algorithms, which often use generic and fixed models.

Adaptability DeepLabCut models can be re-trained and fine-tuned for new datasets or applications, making them adaptable to changing requirements and conditions.

Open source software tools tested here are based on traditional computer vision techniques such as template matching, thresholding, edge detection, and curve fitting. The pros and cons of using simple computer vision and deep learning techniques are detailed in the review by O'Mahony et al. [23]. Briefly, there are several advantages to using deep learning models, such as adaptability to lighting, movement artifacts, using transfer learning approach to train using fewer data points where less data is available, using lightweight architectures such as MobileNet could enable real-time calculations and making algorithms scalable/transferable between applications. Some disadvantages to deep learning models include high computational costs, long training times, inefficient real-time processing in some architectures, and overfitting in some architectures that require careful tuning of hyperparameters. Recently, a web application using novel convolutional neural networks and AdaBelief optimizer was launched, where the user needs to upload data on the website only and will be given the results [25]. This application was based on newer algorithms such as U-Net and achieved accuracy rates above 70-80% [5, 34]. However, the algorithm used a semantic segmentation method that takes into consideration the grayscale values to segment and falls short if the grayscale values are not very different [25].

In conclusion, results show that DeepLabCut, based on MobileNet architecture with a lower number of markers, consumes fewer computational resources during training. Also, the same DeepLabCut architecture with a lower number of markers (2 markers) is more accurate and closer to the values measured by humans than other architectures. This study establishes that with the least amount of training, which spans only an hour, using only a few frames, DeepLabCut can outperform current open-source software and its results are close to the values achieved by a human examiner.

5. Authors' declarations

5.1. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

5.2. Declaration of generative AI in scientific writing

The authors declare that they have not used any type of AI while writing this manuscript

5.3. Data availability

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

5.4. Funding

is work was supported by the Birla Institute of Technology and Science Pilani, Hyderabad Campus intramural funding schemes RIG and ACRG (Grant numbers: 1139 and 1346).

5.5. Author contributions

AB has conducted the experiments, and drafted the manuscript; SM has conducted the experiments and analysed the results; SPK has conceptualized the study, analysed the results, drafted the manuscript and supervised the work. The funding body has no involvement in any of the above-mentioned details.

Acknowledgement

The authors would like to thank Professor Venkateswaran Rajagopalan, Department of Electrical and Electronics Engineering, BITS-Pilani Hyderabad Campus for his insightful comments on the manuscript.

References

[1] A. Biró, A. I. Cuesta-Vargas, J. Martín-Martín, L. Szilágyi, and S. M. Szilágyi. Synthetized multilanguage OCR using CRNN and SVTR models for realtime collaborative tools. *Applied Sciences*, 13(7):4419, 2023. doi:10.3390/app13074419.

[2] F. Boutros, N. Damer, K. Raja, R. Ramachandra, F. Kirchbuchner, et al. Iris and periocular biometrics for head mounted displays: Segmentation, recognition, and synthetic data generation. *Image and Vision Computing*, 104:104007, 2020. doi:10.1016/j.imavis.2020.104007.

[3] Brain Initiative Alliance, Neuroscience. DeepLabCut: Neuroscience research initiative. `https://www.braininitiative.org/toolmakers/resources/deeplabcut/`, [Accessed: 3 Nov 2021].

[4] Y. Chen, M. Adjouadi, C. Han, J. Wang, A. Barreto, et al. A highly accurate and computationally efficient approach for unconstrained iris segmentation. *Image and Vision Computing*, 28(2):261–269, 2010. doi:10.1016/j.imavis.2009.04.017.

[5] W. Chinsatit and T. Saitoh. CNN-based pupil center detection for wearable gaze estimation system. *Applied Computational Intelligence and Soft Computing*, 2017(1):8718956, 2017. doi:10.1155/2017/8718956.

[6] V. D. Costa and P. H. Rudebeck. More than meets the eye: the relationship between pupil size and locus coeruleus activity. *Neuron*, 89(1):8–10, 2016. doi:10.1016/j.neuron.2015.12.031.

[7] W. Fuhl, T. Santini, G. Kasneci, and E. Kasneci. Pupilnet: Convolutional neural networks for robust pupil detection. *arXiv*, 2016. ArXiv:1601.04902. doi:10.48550/arXiv.1601.04902.

[8] S. D. Goldinger and M. H. Papesh. Pupil dilation reflects the creation and retrieval of memories. *Current Directions in Psychological Science*, 21(2):90–95, 2012. doi:10.1177/0963721412436811.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, pp. 770–778. Las Vegas, NV, USA, 27-30 Jun 2016. doi:10.1109/CVPR.2016.90.

[10] H. Himmel and T. E. Faelker. Pupillary function test in rat: Establishment of imaging setup and pharmacological validation within modified Irwin test. *Journal of Pharmacological and Toxicological Methods*, 99:106588, 2019. doi:10.1016/j.vascn.2019.106588.

[11] J. P. Horwath, D. N. Zakharov, R. Mégret, and E. A. Stach. Understanding important features of deep learning models for segmentation of high-resolution transmission electron microscopy images. *npj Computational Materials*, 6(1):108, 2020. doi:10.1038/s41524-020-00363-x.

[12] A. G. Howard. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv*, 2017. ArXiv:1704.04861. doi:10.48550/arXiv.1704.04861.

[13] R. A. Jeyarani and R. Senthilkumar. Eye tracking biomarkers for autism spectrum disorder detection using machine learning and deep learning techniques. *Research in Autism Spectrum Disorders*, 108:102228, 2023. doi:10.1016/j.rasd.2023.102228.

[14] N. Kondo, W. Chinsatit, and T. Saitoh. Pupil center detection for infrared irradiation eye image using CNN. In: *Proc. 2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pp. 100–105. IEEE, Kanazawa, Japan, 19-22 Sep 2017. doi:10.23919/SICE.2017.8105630.

[15] K. Kuraoka and K. Nakamura. Facial temperature and pupil size as indicators of internal state in primates. *Neuroscience Research*, 175:25–37, 2022. doi:10.1016/j.neures.2022.01.002.

[16] R. S. Larsen and J. Waters. Neuromodulatory correlates of pupil dilation. *Frontiers in Neural Circuits*, 12:21, 2018. doi:10.3389/fncir.2018.00021.

[17] M. C. Leong, D. K. Prasad, Y. T. Lee, and F. Lin. Semi-CNN architecture for effective spatio-temporal learning in action recognition. *Applied Sciences*, 10(2):557, 2020. doi:10.3390/app10020557.

[18] M. W. Mathis and A. Mathis. Deep learning tools for the measurement of animal behavior in neuroscience. *Current Opinion in Neurobiology*, 60:1–11, 2020. doi:10.1016/j.conb.2019.10.008.

[19] S. Mathôt. Pupillometry: Psychology, physiology, and function. *Journal of Cognition*, 1(1), 2018. doi:10.5334/joc.18.

[20] P. R. Murphy, R. G. O'Connell, M. O'Sullivan, I. H. Robertson, and J. H. Balsters. Pupil diameter covaries with BOLD activity in human locus coeruleus. *Human brain mapping*, 35(8):4140–4154, 2014. doi:10.1002/hbm.22466.

[21] A. Nasim, A. Maqsood, and T. Saeed. Multicore and GPU based pupillometry using parabolic and elliptic Hough transform. *International Journal of Mechanical Engineering and Robotics Research*, 6(5), 2017. doi:10.18178/ijmerr.6.5.425-433.

[22] T. Nath, A. Mathis, A. C. Chen, A. Patel, M. Bethge, et al. Using DeepLabCut for 3D markerless pose estimation across species and behaviors. *Nature Protocols*, 14(7):2152–2176, 2019. doi:10.1038/s41596-019-0176-0.

[23] N. O'Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, et al. Deep learning vs. traditional computer vision. In: *Advances in Computer Vision: Proc. 2019 Computer Vision Conference (CVC)*, vol. 943 of *Advances in Intelligent Systems and Computing*, pp. 128–144. Springer, LasVegas, USA, 2-3 May 2020. doi:10.1007/978-3-030-17795-9.

[24] M. Privitera, K. D. Ferrari, L. M. von Ziegler, O. Sturman, S. N. Duss, et al. Author correction: A complete pupillometry toolbox for real-time monitoring of locus coeruleus activity in rodents. *Nature Protocols*, 16(8):4108, 2021. doi:10.1038/s41596-021-00493-6.

[25] M. M. Raffaele, F. Carrara, A. Viglione, L. Lupori, L. L. Verde, et al. MEYE: Web-app for translational and real-time pupillometry. *ENEURO*, 8(5), 2021. doi:10.1523/ENEURO.0122-21.2021.

[26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In: *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 4510–4520. Salt Lake City, UT, USA, 18-23 Jun 2018. doi:10.1109/CVPR.2018.00474.

[27] T. Santini, W. Fuhl, and E. Kasneci. Pure: Robust pupil detection for real-time pervasive eye tracking. *Computer Vision and Image Understanding*, 170:40–50, 2018. doi:10.1016/j.cviu.2018.02.002.

[28] T. Satriya, S. Wibirama, and I. Ardiyanto. Robust pupil tracking algorithm based on ellipse fitting. In: *Proc. 2016 International Symposium on Electronics and Smart Devices (ISESD)*, pp. 253–257. IEEE, Bandung, Indonesia, 29-30 Nov 2016. doi:10.1109/ISESD.2016.7886728.

[29] D. Sinha and M. El-Sharkawy. Thin MobileNet: An enhanced MobileNet architecture. In: *Porc. 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pp. 0280–0285. IEEE, New York, NY, USA, 10-12 Oct 2019. doi:10.1109/UEMCON47517.2019.8993089.

[30] S. Sirois and J. Brisson. Pupillometry. *WIREs Cognitive Science*, 5(6):679–692, 2014. doi:10.1002/wcs.1323.

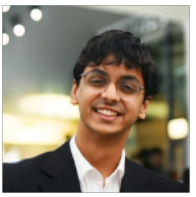
[31] M. E. Suryanto, F. Saputra, K. A. Kurnia, R. D. Vasquez, M. J. M. Roldan, et al. Using DeepLabCut as a real-time and markerless tool for cardiac physiology assessment in zebrafish. *Biology*, 11(8):1243, 2022. doi:10.3390/biology11081243.

[32] Weights & Biases. The AI developer platform to build AI applications and models with confidence. `https://wandb.ai`.

[33] P. Van der Wel and H. Van Steenbergen. Pupil dilation as an index of effort in cognitive control tasks: A review. *Psychonomic Bulletin & Review*, 25:2005–2015, 2018. doi:10.3758/s13423-018-1432-y.

[34] Y.-H. Yiu, M. Aboulatta, T. Raiser, L. Ophey, V. L. Flanagin, et al. DeepVOG: Open-source pupil segmentation and gaze estimation in neuroscience using deep learning. *Journal of Neuroscience Methods*, 324:108307, 2019. doi:10.1016/j.jneumeth.2019.05.016.

[35] B. Zandi and M. Lode. PupilEXT: Flexible open-source platform for high-resolution pupillometry in vision research. *Frontiers in Neuroscience*, 15:676220, 2021. doi:10.3389/fnins.2021.676220.



Amitesh Badkul is currently a Ph.D. student at the Department of Computer Science at the City University of New York. He graduated from BITS-Pilani in 2023 with a Bachelor's degree in EEE and a Master's in Chemistry. His research focuses on developing advanced deep learning models for biological applications, particularly in drug discovery, with an emphasis on uncertainty quantification and out-of-distribution generalization in real-world chemical and biological settings.

Sonakshi Mishra graduated from BITS-Pilani with a Bachelor's degree in ECE and a Master's degree in Mathematics in 2024. She is currently working at Math AI as a software development engineer, focusing on enhancing asset workflows and caching strategies. Her work has achieved a 40% reduction in load times and a 50% improvement in server response times, leveraging Next.js and Node.js for scalable performance.

Srinivasa Prasad Kommajosyula is currently an assistant professor at the Department of Pharmacy at BITS-Pilani Hyderabad Campus. His research interests include hearing loss, depression, anxiety, stroke, and biomedical devices.