computers

MDPI

Article

A Deep Learning Approach for Early Detection of Facial Palsy in Video Using Convolutional Neural Networks: A Computational Study

Anuja Arora [1,*], Jasir Mohammad Zaeem [1], Vibhor Garg [1], Ambikesh Jayal [2] and Zahid Akhtar [3,*]

[1] Jaypee Institute of Information Technology, Noida 201014, India; jasir@zaeem.dev (J.M.Z.); vibhor656@outlook.com (V.G.)
[2] School of Engineering and Technology, CQU University, Brisbane, QLD 883155, Australia; a.jayal@cqu.edu.au
[3] Department of Network and Computer Security, State University of New York Polytechnic Institute, Utica, NY 13502, USA
* Correspondence: anuja.arora29@gmail.com (A.A.); akhtarz@sunypoly.edu (Z.A.)

Abstract: Facial palsy causes the face to droop due to sudden weakness in the muscles on one side of the face. Computer-added assistance systems for the automatic recognition of palsy faces present a promising solution to recognizing the paralysis of faces at an early stage. A few research studies have already been performed to handle this research issue using an automatic deep feature extraction by deep learning approach and handcrafted machine learning approach. This empirical research work designed a multi-model facial palsy framework which is a combination of two convolutional models—a multi-task cascaded convolutional network (MTCNN) for face and landmark detection and a hyperparameter tuned and parametric setting convolution neural network model for facial palsy classification. Using the proposed multi-model facial palsy framework, we presented results on a dataset of YouTube videos featuring patients with palsy. The results indicate that the proposed framework can detect facial palsy efficiently. Furthermore, the achieved accuracy, precision, recall, and F1-score values of the proposed framework for facial palsy detection are 97%, 94%, 90%, and 97%, respectively, for the training dataset. For the validation dataset, the accuracy achieved is 95%, precision is 90%, recall is 75.6%, and F-score is 76%. As a result, this framework can easily be used for facial palsy detection.

Keywords: facial paralysis; landmark; MTCNN; CNN; deep learning

check for updates

Citation: Arora, A.; Zaeem, J.M.; Garg, V.; Jayal, A.; Akhtar, Z. A Deep Learning Approach for Early Detection of Facial Palsy in Video Using Convolutional Neural Networks: A Computational Study. *Computers* **2024**, *13*, 200. https://doi.org/10.3390/computers13080200

Academic Editors: Paolo Bellavista and Wenbing Zhao

Received: 8 June 2024
Revised: 12 July 2024
Accepted: 13 August 2024
Published: 15 August 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Bell's palsy is a peripheral palsy of the facial nerve that results in muscle weakness on one side of the face. It is more common in patients with diabetes. The ordinary techniques for the detection of Bell's palsy such as EMG (electromyography) may be invasive with several after-effects and need extreme sophistication for successful conduction [1]. Thus, there is a need for a non-invasive and quick method to detect facial palsy on faces without a physical intervention/medical equipment process to provide early detection for facial palsy detection. Consequently, facial palsy recognition by a medical expert is not only a time-consuming and labor-intensive procedure but also is dependent on individual expertise for its effectiveness. Thus, the need to develop a method that can help experts scale up and speed up detection is significant itself. Hence, the facial palsy detection problem can clearly benefit from the recent development of image classification and deep learning models. A detailed review report of all the quantification methods from mathematical modeling to deep learning for automatic computer vision-based methods is published to provide a deep understanding of the domain [2]. Another interest review work is published by Boochoon and fellow researchers to present the state of facial palsy assessment, advances through artificial intelligence methods and challenges due to these AI methods [3]. In May 2024,

Nicole Heng and other researchers published a paper where a multimodal fusion-based deep learning model is used for facial palsy detection [4]. Based on these review papers and identifying research gap in existing work, the targeted research aim is to design an artificial intelligence process using an effective deep learning model to identify facial palsy using digital media resources. It is desirable to build an effective facial palsy detection system using a perfectly experimented and suitable deep learning technique which is adaptive to detect palsy in complex environments (in video instead of images, where palsy/without palsy-affected people have different facial expressions). This research paper proposes a different perspective to obtain a solution to the facial palsy detection problem.

The proposed facial palsy detection approach is technically different from existing solutions. In the proposed approach, two convolution neural network variants are applied: the first one is for face and landmark detection, and the second one is for palsy detection. Generally, researchers applied a Haar features-driven cascade face detector, which is considered to be the most appropriate to achieve good face detection performance, but this cascade face detector suffers from degraded performance in real-world applications with variations in lighting and other visual characteristics. Even in case of increasingly complex features, it is harder for these methods to combat environmental conditions and make this task harder. Deep learning advancements with the usage of an MTCNN model have made face detection and face annotation tasks possible and a good fit. Additionally, another convolutional neural network model is designed for facial palsy recognition. This amalgamation of two diverse CNN models is the originality of the present work and achieves excellent performance. The research contribution of the present work is as follows:

- A multitask cascaded convolutional neural network (MTCNN) model is used for face detection and face landmarks detection. The MTCNN extracted landmark features exploit the facial palsy data preparation and rotate face images lying in frames of video data. Further, data augmentation such as rotation of the face to keep the face aligned and extend the dataset for proper training is performed with the help of extracted face landmark features.
- A convolutional neural network model is used to train palsy-affected and unaffected faces for classification as palsy/no palsy. The model for this task is designed based on parametric experimentation and hyperparameter tuning to optimize accuracy and speed.

The remainder of the paper comprises a multi-model facial palsy detection approach which is a combination of two convolutional neural network models. The overview of the MTCNN and CNN is detailed in Section 3. Section 4 discusses the dataset preparation and experimental set up, which is followed by the results in Section 5 where the MTCNN outcome and CNN outcome using multiple performance measures are showcased. Finally, concluding remarks and the future scope are discussed in Section 6.

2. Literature Study

There are a plethora of studies in the literature investigating this area. Hsu et al. demonstrated a Hierarchical Detection Network (HDN) for the detection of facial palsy in 2018 and identified it as the first deep learning-based approach for facial palsy detection [5,6]. Their proposed HDN consists of three components—(1) face detection; (2) landmark detection; and (3) local palsy region detection. Face detection and local palsy detection uses a darknet network (designed to build the well-known YOLO network) with fewer convolution layers and landmark detection using a 3D face alignment network. The dataset is prepared using 32 YouTube videos which consist of a facial expression dataset of 75 patients with and 10 participants without facial palsy. The HDN model achieved a precision of 93% and recall of 88% for facial palsy detection. Storey et al. in 2019 proposed a 3DPalsyNet end-to-end facial palsy detection framework [7]. This model utilizes a 3D convolution neural network with ResNet architecture, and the model achieved an accuracy for facial palsy of 82% and accuracy for mouth motion of 86%. Furthermore, the training time range of 3DPalsyNet to

attain facial palsy detection was 2–4 h depending on frame size ranges 8–16 in duration, so reducing the training and test time without affecting accuracy is troublesome work.

In 2018, Sajid et al. published a comparative analysis of hand-crafted facial palsy features and convolution neural network automatically generated features-driven palsy classification results [8]. In this research work, Sajid et al. introduced a convolutional neural network to generate a discriminant feature, and to prevent overfitting of the CNN model, facial palsy data augmentation is completed using a generative adversarial network deep learning model [9–11]. This work attained accuracy of 93% using VGG-16, a pre-trained CNN model on an originally set facial palsy scale by the House and Brackmann (HB) [12]. The authors compared results with two handcrafted facial palsy classification works [13,14] using machine learning classification models—support vector machine (SVM) [13,14] and a latent Dirichlet allocation (LDA) algorithm [13]. The deep learning model achieved an outcome that is much better than the outcome of hand-crafted features and the machine learning model's outcome. Facial palsy classification performance using VGG-16 is 93%. A recently published work in May 2024 showcased the results of facial palsy detection using numerous pre-trained models including ResNet, AlexNet, DenseNet, GoogleNet, and VGG16 to detect facial palsy in real time and achieved 98% accuracy [15].

A case study of a healthy and CFP patient is considered for central facial palsy detection in case of medical emergencies, and a subjective visual assessment is performed using two evaluation indices—symmetry of mouth and difference in mouth shape with accuracy of 84.7% [16]. Another different research direction to handle facial palsy detection and classification work was published by Jiang et al. [17]. In their mentioned computational image analysis, the facial blood of facial palsy patients is taken into consideration using a laser speckle contrast imaging technique to generate an RGB image and blood flow image. Further, an enhanced segmentation approach was taken for palsy region extraction, and the HB score was quantified. Segmentation evaluation is performed using a Dice coefficient [18]. Jiang et al. stated that the neural network achieved optimal HB classification accuracy as compared to SVM, and the worst performance is achieved by the K-NN model. After studying the entire literature, the authors observed that considering face data for facial paralysis analysis is commonly used in the literature where the result varies, and the accuracy deviates according to face orientation in the taken dataset image. Sometimes, face orientation in an image can match the facial paralysis image. To enrich the training of a deep learning model, extension of the facial palsy dataset is must. Hence, data augmentation is performed for deep learning image classification performance improvement [9]. The authors of this research paper discuss and compare the different types of image augmentation methods to solve this problem. The data augmentation methods such as image rotating, cropping, zooming, histogram-based methods and style transfer and generative adversarial networks are detailed. The authors also present their own method of data augmentation based on image style transfer that allows the generation of new images of high perceptual quality that combine the content of a base image with the appearance of other ones. Hence, the newly created images can be used to pre-train the given neural network to improve the training process efficiency [9].

3. Multi-Model Facial Palsy Detection Framework

This section discusses the multi-model framework in which one variant of the CNN model is used for face extraction and landmark detection from video frames and another variant of CNN is used for facial palsy detection. Both models are implemented for different perspectives and utilizing distinct deep learning convolution models.

3.1. Multi-Task Cascaded Convolutional Network

A facial paralysis model requires an approach to extract faces from the taken video dataset. Therefore, the process begins by detecting the face present in the input and its associated landmarks on it using an appropriate and effective deep learning model. For this same, a multi-task cascaded convolutional network (MTCNN) [19] model is taken

into consideration, which has been used in the literature for face detection using landmark features. Primarily, the MTCNN model was introduced by Zhang et al. in 2016 for joint face detection and alignment, which is a cascade framework of three convolutional neural network models—Proposal Network (P-Net), Refine Network (R-Net), and Output Network (O-Net). The input for these three networks (P-Net, R-Net, and O-Net) is the same image using different sizes; hence, we begin with resizing of the image at different scales. These images are input for three cascaded networks. First, P-Net is a fully connected convolution network (FCN) which is used to take candidate windows and predict the localization of boxes to detect a face as an object using a bounding box regression vector. The P-Net architecture taken from Zhang et al. (2016) is shown in Figure 1. The outcome of P-Net is candidate windows where highly overlapped candidates are merged. This outcome is fed into R-Net (Refine Network), which is a convolution neural network that consists of a dense layer at the end of the R-Net architecture. The outcome of R-Net after reducing the candidates is a bounding box of a face having four element vectors and facial landmark localization with 10 element vectors (see Figure 1).

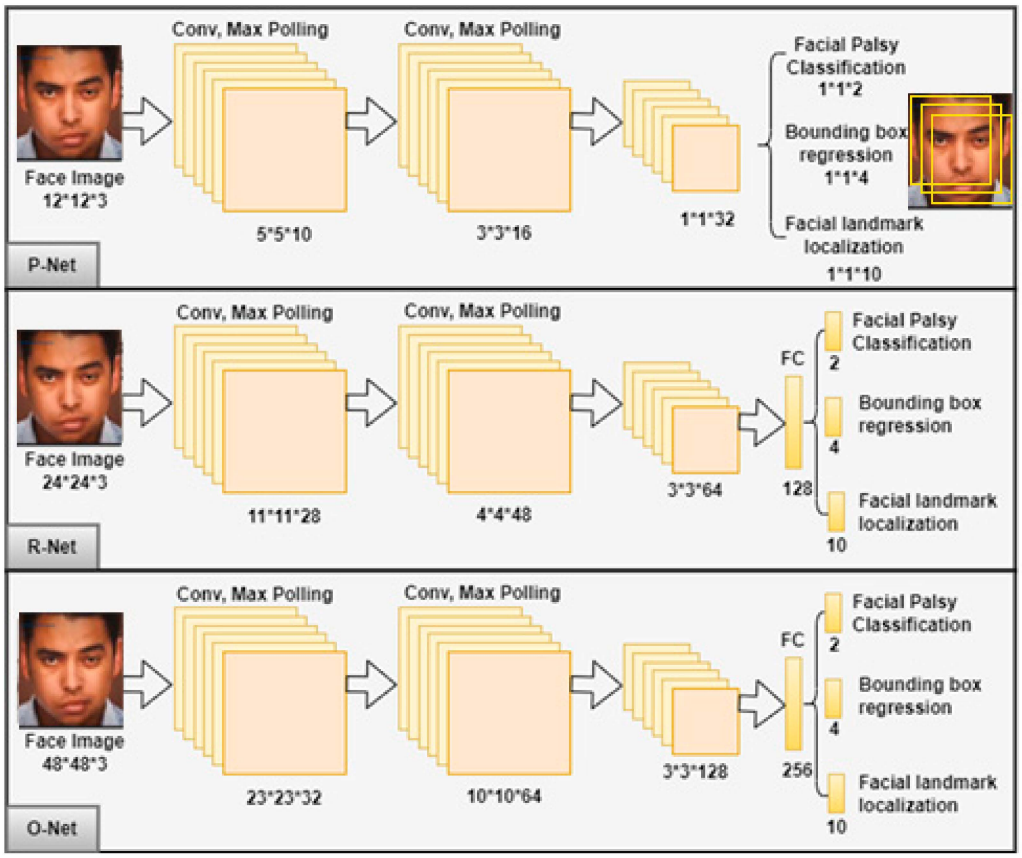


Figure 1. MTCNN architecture [adapted from [19], original MTCNN research paper].

Last, there is the output network (O-Net), which is again a convolution neural network that gives details of faces, and the output is three facial landmark positions in the considered problem (one landmark on the nose and two landmarks on the mouth). In our case, the output network (O-Net) facial landmark localization is 3 (three landmark features instead of the 10 mentioned in Figure 1). Hence, the MTCNN is a cascade framework of three CNNs which efficiently detect and landmark faces. These landmarks are further used to align the face such that the eyes lie on the horizontal. After the data augmentation for alignment, the output is again passed through the MTCNN to detect the new bounding box for the aligned face; i.e., the MTCNN was executed two times on the considered dataset. This method can also be used to aid in quickly generating facial palsy datasets to train the final classifier along with manual labelling. Furthermore, a self-disciplined CNN is

exercised on detected faces along with landmark points to classify face palsy (affected and not affected).

3.2. Convolutional Neural Network for Facial Classification

Here, a convolutional neural network (CNN) is used to classify whether the processed input face is having facial palsy or not. CNN basically refers to neural networks that apply convolution technique on images [20–22]. The model for this task is designed based on parametric experimentation and hyperparameter tuning to optimize accuracy and speed [23]. Finally, the system processes videos of people (speaking) to detect whether they are affected by palsy. Each frame of the considered input video stream is processed to extract an aligned face and then classify whether it shows signs of palsy. The video stream frames yield better palsy detection accuracy as compared to image collection. A convolutional neural network model containing 11 layers (8 CNN layers and 3 fully connected layers) is used after a detailed hyperparameter tuning experimentation to achieve the best accuracy. The architecture of the applied convolutional neural network is shown in Figure 2.

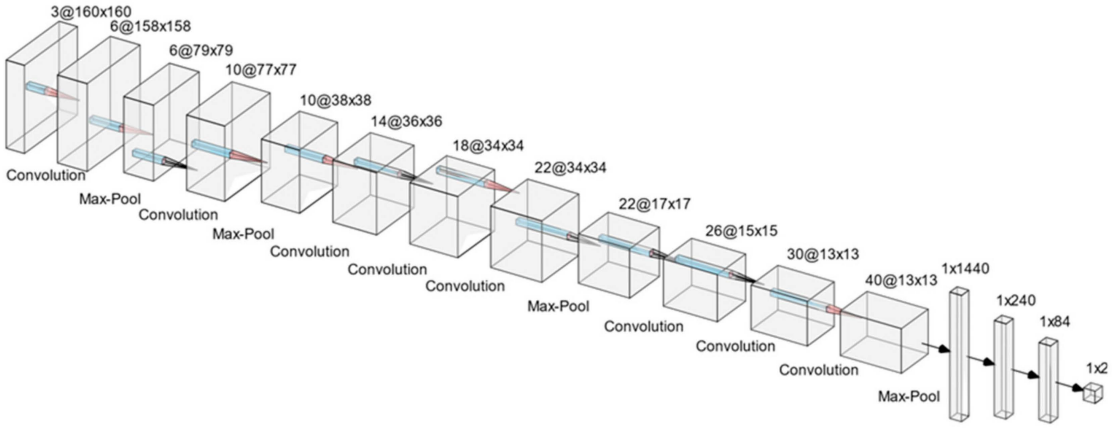


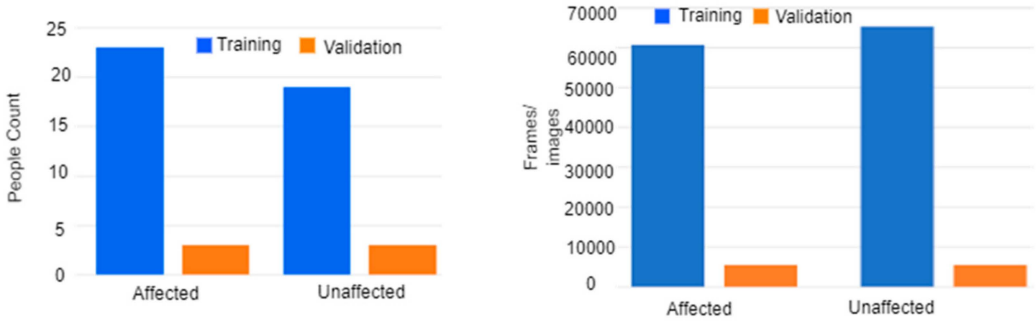
Figure 2. Convolution model for facial palsy detection, the output activation maps of each layer are labeled (figure of proposed CNN model is designed using NN-SVG).

The number of convolutional layers and max pooling layers are decided after parametric experimentation. The model takes three-channel RGB images as input of size 160 × 160. When transforming images into tensors, their values are converted to floats between 0 and 1 instead of integers between 0 and 255 to improve performance on GPUs and maintain numerical stability. The labels on the diagrams are the activation output. In the first step, there are 6, 3 × 3 kernels, which result in an output of 6 × 158 × 158 from the initial input of 3 × 160 × 160 and then so on. The CNN classification model's layer dimensions are height × width × F, where F is the number of filters in each layer. Filters are a matrix of n × n dimensions to detect a specific feature in an image such as edge detection, contrast, correction, etc. Here, the model consists of 397,182 trainable features. Although we are applying filters and max-pooling results in the detection of features in a linear state, to make our model robust against non-linearity, a rectified linear unit (ReLU) is used. Rectified linear units go through the image pixel by pixel and change any value less than zero to zero, and any value above zero remains as it is. Pooling is used to downsample the input; for example, in 2 × 2 pooling with stride 2, a filter is applied on the complete matrix step by step and reduces these four pixels to one: this combined with the stride reduces the size by one fourth by halving the input channels in both dimensions. In our case, max pooling is used: that means, of those four pixels, the one with the maximum value was retained in the output, and the rest are discarded. The two main advantages of pooling are reduction in the dimensions while preserving the special invariance.

4. Dataset Preparation and Experimental Setup

A few datasets and tools to prepare features of facial palsy are already available [24]. But the purpose of the study is to work on a real-time dataset. Henceforth, the dataset is self-prepared from the videos of the people with/without facial palsy from different YouTube creators. A python script is written to process videos in batches of frames and extract the faces in those frames into sets of images. Also, the idea of using videos instead of photos was adapted to avoid the error of wrong classification based on several factors such as insufficient data, person is talking/having a contorted facial expression, etc. In contrast, considering frames of a talking person gives the most natural, real and importantly enough photos to analyze and predict.

The training and test dataset distribution according to people count and frames is shown in Figure 3a,b and Table 1. In total, 23 people with Bell's palsy and 19 people without Bell's palsy videos were taken into consideration, which consist of 66,607 frames for palsy-affected persons and 70,585 frames for unaffected persons in the complete dataset. The data were further divided into two parts: the training set and validation set, where the model was trained on the training dataset, and the validation set was kept aside for prediction by the trained model.



(**a**) People distribution (**b**) Image/Frames distribution

Figure 3. Training and validation dataset partition.

Table 1. Training and validation dataset statistics.

Training Dataset	Validation Dataset
23 People—Affected	3 People—Affected
19 People—Unaffected	3 People—Unaffected
125,825 frames	10,800 Frames

Experiments are performed using the python programming language to validate the proposed facial palsy detection process. The number of python libraries that have been used is detailed in Table 2.

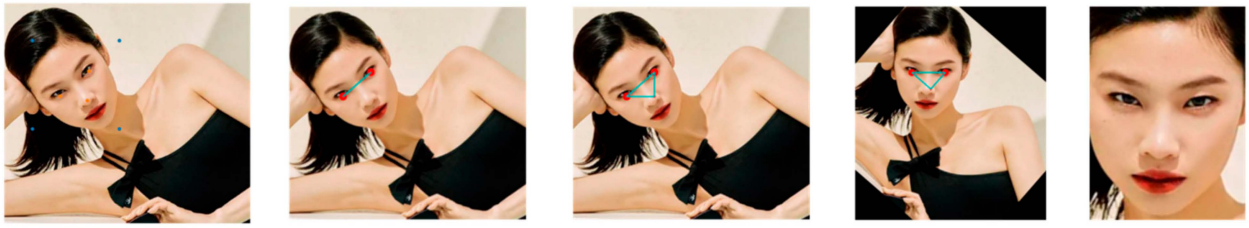
Table 2. Experimental setup.

Resource	Details
Python	Programming language
OpenCV	To read, process and write image and video data
PyTorch	Used as the main machine learning library to implement models and train
Facenet-pytorch	A pretrained implementation of MTCNN
Scikit-learn	Metrics library to evaluate performance of trained models
Matplotlib	Used to plot data for analysis and presentation in report

5. Results and Evaluation

5.1. Face Detection, Extraction, and Augmentation Outcome

As discussed above, the MTCNN model is used in order to detect and extract faces. This process is applied on all the video frames/images of the considered videos. The MTCNN detects faces and helps to mark three landmarks (both eyes and nose) on faces, as shown in Figure 4. These landmarks are used to determine whether the face is straight or at an angle. The incline of the image was determined by the vertical difference in the position of the eyes. Once the angle of incline was determined, the image was rotated about the nose by that specific angle so that the faces given to the models for the training and inference are aligned and consistent. Once it was ascertained that the face in the image is straight, the face in the image was again detected and then extracted, that is, cropped to that specific ratio so that only the face was left in the image.



(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 4. The MTCNN results for face and landmark detection, face enhancement (https://www.thestar.com.my/lifestyle/entertainment/2021/09/17/model-jung-ho-yeon-goes-from-ny-fashion-week-to-making-her-acting-debut-in-new-series-squid-game, accessed on 1 December 2021). (**a**) Face and Landmark detection. (**b**) Calculate eye angle. (**c**) Measure angle to rotate image. (**d**) rotated image. (**e**) Extract Image.

Hence, the first step basically consisted of finding the face using the MTCNN and checking if the face was straight or not. Further, if the face is not aligned, then aligning it, once again identifying the face, and finally extracting the face and saving it. The complete process outcome on an image is shown in Figure 4a–e.

5.2. Facial Palsy Detection Result

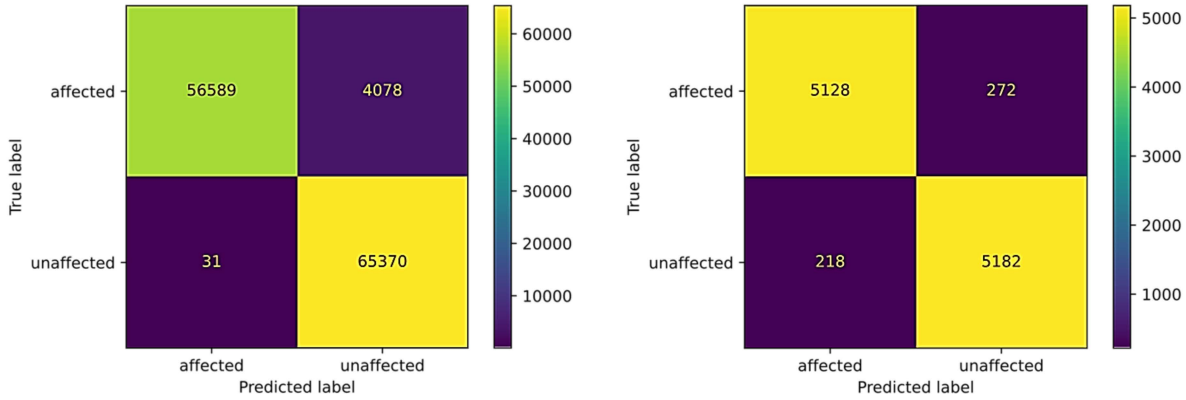
Another CNN model is used to achieve the defined objective of facial palsy detection in each image of a face. The inference python script gives the user an option to directly run the model on the stored test video data directly from the command line interface. The model runs on the provided test data, and a classification is made by the defined CNN model (Section 3.2). Finally, based on inference of the trained/learned model, a bounding box is drawn on the face of the person in the provided data. A red box means the presence of Bell's palsy is detected by the model, and a green box is around the face otherwise. A sample output is shown in Figure 5.

The confusion matrix of the applied CNN-attained outcome after 10 epochs is shown in Figure 6, which shows the actual and false labels count of palsy-affected and unaffected cases. These results are on the best parameters' setting to obtain the best attained accuracy and minimum false positive rate. In the training data, 4109 are wrongly classified/detected classes, which is around 3% (4109 out of 125,825) wrong classification in the training set and 490, i.e., 4.5% (490 out of 10,800) wrong classification in the validation set.

The accuracy, precision, recall, and F-score plots of each epoch are depicted in Figures 7–10, respectively. Figure 7a,b show the training and test run accuracy on the training and validation set. It is observed that the maximum accuracy achieved was 95.90% (Figure 7b in validation outcome).

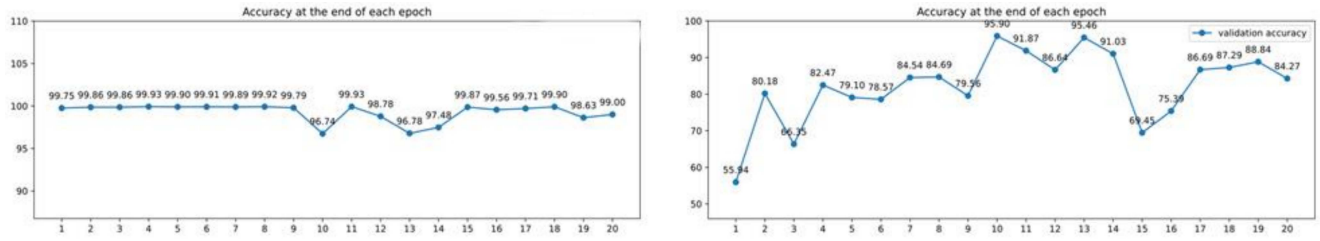


Figure 5. Facial palsy affected and not affected classification outcome (https://www.youtube.com/watch?v=1weQBIGTACo (accessed on 1 December 2021) [permission graded by YouTube Video Owner]).



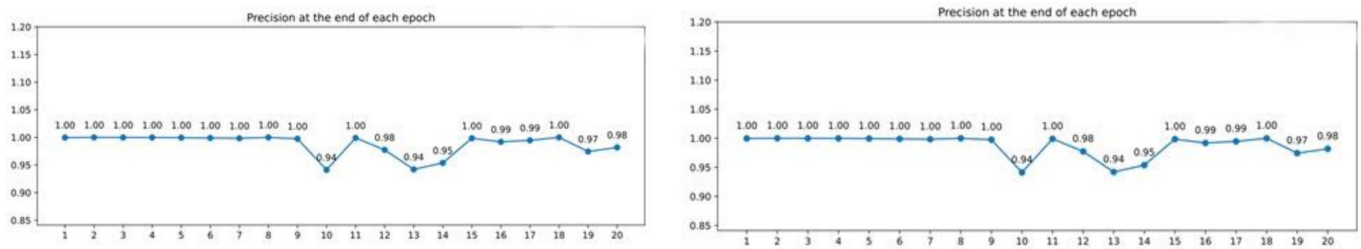
(**a**) Training Set (**b**) Validation Set

Figure 6. Confusion matrix of the chosen model on the training set.



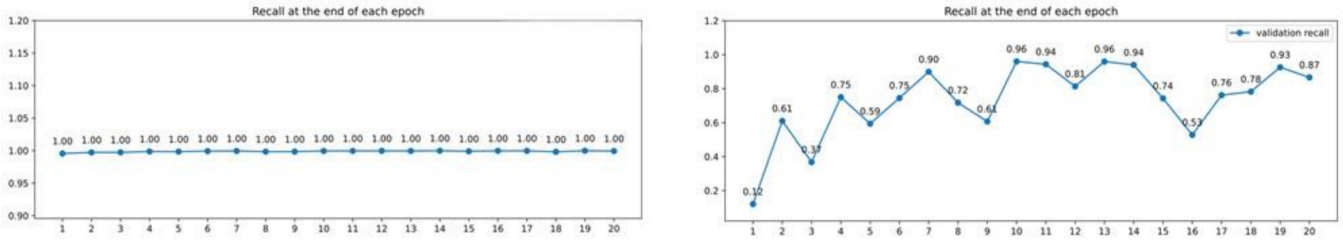
(**a**) Training Accuracy (**b**) Validation Accuracy

Figure 7. Training and validation accuracy plot of Palsy detection using proposed approach.



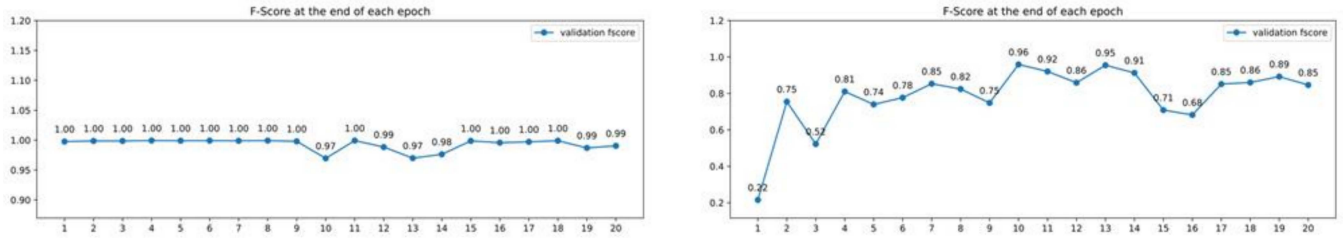
(**a**) Training Precision (**b**) Validation Precision

Figure 8. Training and validation precision plot of Palsy detection using proposed approach.



(**a**) Training Recall (**b**) Validation Recall

Figure 9. Training and validation recall plot of Palsy detection using proposed approach.



(**a**) Training Precision (**b**) Validation Precision

Figure 10. Training and validation F-score plot of Palsy detection using proposed approach.

Figure 8 shows the precision and Figure 9 shows the recall plot of the training and validation results, where the precision and recall are in the range of 94–100% in the training data, whereas the average precision is 90% and the average recall is 75.6% for the validation set.

Figure 10 discusses the F-score of the training and validation set, where the proposed facial palsy approach is able to obtain approximately 97% in the training set and 76% validation accuracy in 20 epochs.

The validation set comprised six videos (three facial palsy affected and three facial palsy unaffected) of 1 min each, and the model was finally run on these videos to obtain the overall chances of Bell's palsy as the model is meant to infer from a video stream.

The overall average performance outcome of accuracy, precision, recall and F-score for training and validation dataset are shown in Table 3.

Table 3. Facial palsy performance outcome.

Dataset	Accuracy	Precision	Recall	F-Measure
Training	99.2%	98%	1	98%
Validation	82.2%	98%	73.2%	78.4%

5.3. Ablation Study: Frames Study

The percentage of frames with the presence of palsy compared to frames of those not having palsy is shown in Table 4 for facial palsy-affected videos. The proposed model outperformed and detected a minimum of 86% of the frames of faces with palsy. Table 5 shows the presence of palsy in unaffected palsy videos, and the results show a 9% maximum gained error.

Table 4. Facial palsy affected validation set outcome details.

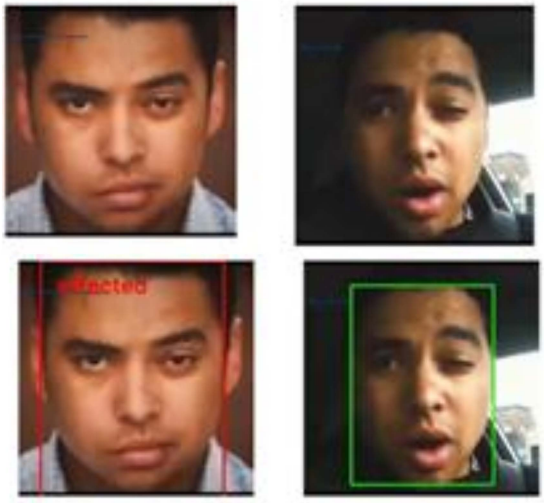
Video Name	Number of Frames in Which a Face Was Detected	Number of Frames in Which the Model Detected Palsy	Percentage of Frames with Presence of Palsy Compared to Frames with Faced
affected/1.mp4	1800	1550	86.11%
affected/2.mp4	1800	1799	99.94%
affected/3.mp4	1800	1786	99.22%

Table 5. Facial palsy unaffected validation set outcome details.

Video Name	Number of Frames in Which a Face Was Detected	Number of Frames in Which the Model Detected Palsy	Percentage of Frames with Presence of Palsy Compared to Frames with Faced
unaffected/1.mp4	1800	164	9.11%
unaffected/2.mp4	1800	81	4.5%
unaffected/3.mp4	1800	32	1.72%

5.4. Ablation Study: Validation Set Test Cases

A few test cases are placed here to showcase the results of the proposed facial palsy detection model. To build a robust and accurate system, the proposed methodology is validated on the video stream dataset where palsy affected/unaffected persons are detected in the videos while talking to train the system. Figure 11 shows the validation test cases of pass and fail test cases. Figure 11a shows the facial palsy-affected test cases that the implemented system can identify correctly, whereas Figure 11b shows the faces having palsy that the system is not able to recognize and assigns to the wrong class.



(**a**) (**b**)

Figure 11. Test cases of validation dataset facial palsy-affected/unaffected classification. (**a**) Affected, Test Case: Pass. (**b**) Affected, Test Case: Fail.

5.5. Limitation

This research work was performed on a limited set of video data to validate the efficacy of the proposed multi-model facial palsy detection framework. The results are validated on a synthetic dataset and both convolutional neural network models—MTCNN for face and landmark detection and CNN for palsy detection—worked perfectly well. The achieved accuracy is also almost the same as mentioned in Table 3 for these videos. Model overfitting behavior has been tested for this synthesis dataset as well, and no issue was faced. These

results are not placed due to data privacy in the paper. The future aim is to extend this work and check the palsy detection result on large amount of open data.

6. Conclusions and Future Scope

An efficient and precise multi-model facial palsy detection framework is established. This proposed multi-model is able to detect facial palsy by using deep features using two variants of convolutional neural network models. First, a multi-task cascaded convolution neural network model (MTCNN) used varying sizes of the same input face image to detect faces and their landmarks (in this case, the eyes and the nose). The MTCNN itself is a combination of three neural network models—P-Net, R-Net, and O-Net, which can detect face and landmark points on the face. These landmark points are used for the preprocessing of faces and to create the dataset. Furthermore, an eight-layer convolutional neural network model is then employed to classify images as facial palsy/no facial palsy. The complete process is validated on videos to make it more robust and effective for the detection of palsy in complex environments. The system achieved excellent performance in facial palsy detection. In future, we intend to focus on the following:

- To detect palsy in videos, preprocessing is essential to classify the condition in individual subjects. The selection of shots and frames must be dynamically tailored to target each individual accurately.
- Generate and share enriched video datasets of facial palsy as open data in repositories to support extensive research initiatives.

Author Contributions: Conceptualization, A.A., A.J. and Z.A.; methodology, J.M.Z. and V.G.; validation, A.A., J.M.Z. and V.G.; writing—review and editing, Z.A. and A.J.; project administration, A.A., A.J. and Z.A.; funding acquisition, Z.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: https://huggingface.co/datasets/jasir/palsynet-data (accessed on 1 August 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Guntinas-Lichius, O.; Volk, G.F.; Olsen, K.D.; Mäkitie, A.A.; Silver, C.E.; Zafereo, M.E.; Ferlito, A. Facial nerve electrodiagnostics for patients with facial palsy: A clinical practice guideline. *Eur. Arch. Otorhinolaryngol.* **2020**, *277*, 1855–1874. [CrossRef] [PubMed]
2. Vrochidou, E.; Papić, V.; Kalampokas, T.; Papakostas, G.A. Automatic Facial Palsy Detection—From Mathematical Modeling to Deep Learning. *Axioms* **2023**, *12*, 1091. [CrossRef]
3. Boochoon, K.; Mottaghi, A.; Aziz, A.; Pepper, J.P. Deep Learning for the Assessment of Facial Nerve Palsy: Opportunities and Challenges. *Facial Plast. Surg.* **2023**, *39*, 508–511. [CrossRef] [PubMed]
4. Oo, N.H.Y.; Lee, M.H.; Lim, J.H. Exploring a Multimodal Fusion-Based Deep Learning Network for Detecting Facial Palsy. *arXiv* **2024**, arXiv:2405.16496.
5. Tiemstra, J.D.; Khatkhate, N. Bell's Palsy: Diagnosis and Management. *Am. Fam. Physician* **2007**, *76*, 997–1002. [PubMed]
6. Hsu, G.S.J.; Huang, W.F.; Kang, J.H. Hierarchical Network for Facial Palsy Detection. In Proceedings of the CVPR Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 580–586.
7. Storey, G.; Jiang, R.; Keogh, S.; Bouridane, A.; Li, C.T. 3DPalsyNet: A facial palsy grading and motion recognition framework using fully 3D convolutional neural networks. *IEEE Access* **2019**, *7*, 121655–121664. [CrossRef]
8. Sajid, M.; Shafique, T.; Baig, M.J.A.; Riaz, I.; Amin, S.; Manzoor, S. Automatic grading of palsy using asymmetrical facial features: A study complemented by new solutions. *Symmetry* **2018**, *10*, 242. [CrossRef]
9. Mikołajczyk, A.; Grochowski, M. Data augmentation for improving deep learning in image classification problem. In Proceedings of the 2018 International Interdisciplinary PhD Workshop (IIPhDW), Swinoujscie, Poland, 9–12 May 2018; pp. 117–122.
10. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
11. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2672–2680.
12. Evans, R.A.; Harries, M.L.; Baguley, D.M.; Moffat, D.A. Reliability of the House and Brackmann grading system for facial palsy. *J. Laryngol. Otol.* **1989**, *103*, 1045–1046. [CrossRef] [PubMed]

13. Kim, H.S.; Kim, S.Y.; Kim, Y.H.; Park, K.S. A smartphone-based automatic diagnosis system for facial nerve palsy. *Sensors* **2015**, *15*, 26756–26768. [CrossRef] [PubMed]
14. Song, I.; Yen, N.Y.; Vong, J.; Diederich, J.; Yellowlees, P. Profiling bell's palsy based on House-Brackmann score. In Proceedings of the 2013 IEEE Symposium on Computational Intelligence in Healthcare and E-Health (CICARE), Singapore, 16–19 April 2013; pp. 1–6.
15. Amsalam, A.S.; Al-Naji, A.; Daeef, A.Y. Facial palsy detection using pre-trained deep learning models: A comparative study. In Proceedings of the AIP Conference Proceedings, Jaipur, India, 22–23 May 2024; Volume 3097.
16. Ikezawa, N.; Okamoto, T.; Yoshida, Y.; Kurihara, S.; Takahashi, N.; Nakada, T.A.; Haneishi, H. Toward an application of automatic evaluation system for central facial palsy using two simple evaluation indices in emergency medicine. *Sci. Rep.* **2024**, *14*, 3429. [CrossRef] [PubMed]
17. Jiang, C.; Wu, J.; Zhong, W.; Wei, M.; Tong, J.; Yu, H.; Wang, L. Automatic facial paralysis assessment via computational image analysis. *J. Healthc. Eng.* **2020**, *2020*, 2398542. [CrossRef] [PubMed]
18. Dice, L.R. Measures of the amount of ecological association between species. *Ecology* **1945**, *26*, 297–302. [CrossRef]
19. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Process. Lett.* **2016**, *23*, 1499–1503. [CrossRef]
20. Arora, A.; Taneja, A.; Gupta, M.; Mittal, P. Virtual Personal Trainer: Fitness Video Reognition Using Convolution Neural Network and Bidirectional LSTM. *IJKSS* **2021**, *12*, 1–21. [CrossRef]
21. Arora, A.; Jayal, A.; Gupta, M.; Mittal, P.; Satapathy, S.C. Brain tumor segmentation of mri images using processed image driven u-net architecture. *Computers* **2021**, *10*, 139. [CrossRef]
22. Ramprasath, M.; Anand, M.V.; Hariharan, S. Image classification using convolutional neural networks. *Int. J. Pure Appl. Math.* **2018**, *119*, 1307–1319.
23. Sharma, N.; Jain, V.; Mishra, A. An analysis of convolutional neural networks for image classification. *Procedia Comput. Sci.* **2018**, *132*, 377–384. [CrossRef]
24. Arora, A.; Sinha, A.; Bhansali, K.; Goel, R.; Sharma, I.; Jayal, A. SVM and Logistic Regression for Facial Palsy Detection Utilizing Facial Landmark Features. In Proceedings of the 2022 Fourteenth International Conference on Contemporary Computing, Noida, India, 4–6 August 2022; pp. 43–48.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.