FACE LANDMARKS DETECTION OPEN CV WITH PYTHON

B.SURYANARAYANA MURTHY, KADALI PRASANNA

[1] **Assistant Professor MCA DEPT,** Dantuluri Narayana Raju College **,Bhimavaram, Andhrapradesh**

Email id: - suryanarayanamurthy.b@gmail.com

[2]**PG Student of MCA,** Dantuluri Narayana Raju College, **Bhimavaram, Andhrapradesh**

Email id: - Prasannakadali777@gmail.com

ABSTRACT

Facial landmarks are distinctive points in human faces that are used for a variety of tasks such as facial expression analysis, lip reading or face recognition. The performance on these tasks depends heavily on the accuracy of the detected facial landmarks Facial landmarks are distinctive points in human faces that are used for a variety of tasks. It is challenging to accurately locate facial landmarks even on faces that are partially occluded by glasses, facial hair or other objects. In this work we introduce a new approach to tackle these challenges on unconstrained frontal and semi-frontal face images. The proposed solution is a new deep learning based algorithm that is built on the Stacked Hourglass Network which has proven to be effective for human pose estimation, a task similar to facial landmark detection. The algorithm processes face images by repeatedly down- and up sampling the image and thus analyzes it on multiple scales. The Stacked Hourglass Network is trained using Wing loss and regresses coordinates using a Differentiable Spatial To Numerical Transform. Our algorithm is able to outperform current state-of-the-art solutions on the 300-W and Menpo datasets in terms of the point-to-point normalized error. Additionally, a neural Point Distribution Model is employed as a shape model that refines the predictions made by the Stacked Hourglass Network. By adding the Point Distribution Model, the prediction error on the inner facial landmarks of the challenging test set of 300-W reduces even more. The Point Distribution Model achieves the biggest improvements on the inner landmarks of faces with strong head poses while improving the predictions of landmarks on the outline is more challenging.

1. INTRODUCTION

In order to achieve satisfactory human-computer interaction it is crucial for computers to understand the content of images. Analyzing images of human faces is an active research field with numerous different subtopics. Some focus on face recognition [42, 64, 75] which can be used to unlock a phone or open a door by looking at a camera. Analyzing facial expressions and emotions [23, 70] is another direction that can be useful for medical applications or to detect if a person is stressed or relaxed and muting or enabling notifications over new messages based on this. Furthermore, estimating a person's head pose [46, 52, 58, 59] or gaze direction [61, 62] can be used to understand conversations or to detect whether a car driver is focused or distracted.

Another research direction focuses on detecting facial landmarks which are distinctive points with a semantic meaning that are distributed over a human's face, for example around the eyes, mouth and outline. Figure 1.1 shows three faces with 68 annotated facial landmarks. Facial landmarks serve as input to other tasks such as lip reading [15, 43] to improve speech recognition accuracy, facial expression analysis, emotion detection or face recognition [44, 64].

The problem of facial landmark detection is often called facial alignment or facial landmark localization. It can be formally described as finding the (x, y) coordinates of $l \in N+$ semantic points on an image of a human face. Hence, the output of a facial landmark detection algorithm is a vector p = [x1, y1, ..., xl, yl] ∈ R2l. There is also 3D facial landmark detection [49, 80], but this work focuses on 2D coordinates.

Various challenges arise when designing a facial landmark detection algorithm. For example, images can be of varying quality due to camera resolution, illumination or distance between the camera and the face. Moreover, faces can be occluded by hats, glasses or facial hair. Furthermore, faces appear differently depending on the Facial expression.

Finally, the face can be shown in a non-frontal position and there is natural variance due to different ethnicity, gender, age and individual appearance. We focus on semi-frontal faces which include faces viewed from different angles as long as no landmark is occluded by the face itself. Hence, we exclude profile faces from our work. The images are allowed to show faces in uncontrolled settings such as indoor or outdoor, under varying light conditions and with arbitrary backgrounds.

In the early years of computer vision most approaches to analyze images made extensive use of hand-crafted features such as the scale invariant feature transform (SIFT) [40], local binary patterns (LBP) [1] or histogram of oriented gradients (HOG) [20]. These features were used to train classic machine learning algorithms like support vector machines (SVM) [7] or random forests [8].

In 2012, Krizhevsky et al. proposed AlexNet [35], a convolutional neural network that was able to beat the state-of-the-art in image classification by a large margin. Inspired by this success, many researchers in the computer vision domain started to use neural networks and were able to improve their results. The advantage of using (convolutional) neural networks is that no hand-crafted features are needed. Moreover, neural networks are trained to extract hierarchies of complex features by optimizing a loss function using back-propagation. Those features are optimized for the problem at hand, which is the main reason why they outperform hand-crafted features.

2. LITERATURE SURVEY AND RELATED WORK

In this chapter an overview over relevant publications in the field of facial landmark detection is presented. We first start with classic approaches that allowed for first successes in facial landmark detection in Section 2.1. Similar to other computer vision problems, the rise of deep learning models has enabled more powerful algorithms in the facial landmark domain as well. A selection of deep learning based solutions is presented in Section 2.2. Finally, the baselines we use to compare our models to are presented in Section 2.3.

2.1. Classic approaches for facial landmark detection

Most algorithms for facial landmark detection can be classified as either model-based or regression-based [79]. They differ in how the coordinate prediction is done.

Model-based approaches learn constraints on the arrangement of landmarks relatively to each other. This is useful to locate landmarks in non-rigid objects since the locations of landmarks are often constrained by other landmarks. We present three important model-based algorithms: Active Shape Model (ASM) in Section 2.1.1, Active Appearance Model (AAM) in Section 2.1.2 and Constrained Local Model (CLM) in Section 2.1.3.

Regression-based approaches do not have an implicit shape model. They instead operate directly on the image and regress the coordinates of the landmarks. Cascaded regression approaches run multiple regression models consecutively, each one refining the predictions of its predecessor. An overview over regression-based algorithms as well as cascaded regression solutions is presented in Section 2.1.4.

Some of the most important classic approaches are presented in this section. However, there are many more algorithms for facial landmark detection. Wang et al. [71] give a comprehensive survey on facial landmark detection algorithms that were published before 2014.

2.1.1 Active Shape Models (ASMs)

In 1995, Cootes et al. [17] published ASMs, an algorithm that puts constraints on the locations of individual landmarks. Possible shape variations are learned from the training set and stored in a Point Distribution Model (PDM). Before learning the variations, all samples in the training set are aligned by rotating, scaling and translating them so that the difference between samples is minimized. The idea is to have all landmarks in all samples approximately at the same position so that the actual shape variations can be estimated without the influence of pose variations. The PDM stores the average locations of each landmark and the main modes of variation, including correlations of shape parameters. The PDM finds a basis of the shape parameters where the shape parameters are uncorrelated, allowing to change each of the parameters and still generating a valid shape. This is done using Principal Component Analysis (PCA). Once learned, the model can be used to generate new shapes that were never seen in training by varying the uncorrelated shape parameters.

The ASM uses the PDM to find the landmarks in an unseen test image. Locating land- marks involves finding shape parameters that correspond to a shape as close as possible to the test image. The original algorithm assumes that landmarks lie on strong edges. An ASM does not model texture but only looks for edges in arbitrary textures (around the current estimate). The initial location estimates are set to be the mean of the training set. Then, iteratively a local region around each landmark is analyzed by looking on the normal through each landmark and finding the strongest nearby edge. This is then used to update the pose and shape parameters to best fit the new landmark locations. Important to note is that the estimates are not updated directly. Rather, they are updated by finding shape parameters that map to locations that are as close as possible to the new estimate. This is important as otherwise the shape constraints could be violated. The whole process is repeated until convergence.

The algorithm can be improved in terms of speed and accuracy by following a coarse-to- fine approach. That means that the algorithm is first executed on a blurred image. The obtained locations are then used as the initial estimate when running the algorithm on a finer resolution. The process is repeated until the finest resolution is reached.

Although the ASM paper used resistors and human hands as example objects, the algorithm was also an important milestone in the history of facial landmark detections since it allowed to model non-rigid shapes [18].

2.1.2 Active Appearance Models (AAMs)

Building on the success of ASMs, Cootes et al. [16] introduce AAMs. The main difference to ASMs is that texture variation around the landmarks is learned in addition to the shape variation. The AAM can be used to synthesize a whole image using the shape and appearance parameters, hence it is a generative model. Landmarks in unseen images are located by finding shape and appearance parameters that minimize the difference between the generated image and the test image.

The shape model is built in the same way as in ASMs. The appearance model learns the appearance variations in shape-free images. A shape-free image is obtained by aligning the original image with the mean image. To generate a synthetic image, first the appearance parameters are used to generate an image that is shape-free. Then the shape parameters are used to warp it to the desired shape using triangulation-based interpolation.

To find the optimal shape and appearance parameters, the error between the original image and the synthesized image is minimized. In comparison, the ASM is not minimizing distances between images but seeks to improve the current estimate by searching around a small local area. However, the AAM is a holistic model.

Similar to ASMs, the authors construct a multi-resolution model on a Gaussian image pyramid. For each level on the pyramid there is a separate model. Working from coarse to fine, the algorithm is both faster and more accurate than using only the finest resolution.

The algorithm is applied to the problem of facial landmark detection in the original AAM paper. According to [18], ASMs are more accurate and faster than AAMs, but the latter are able to better match the texture. According to [21] AAMs suffer from lightning changes and bias towards the mean face.

2.1.3 Constrained local models (CLMs)

Another class of shape models are CLMs, proposed by Cristinacce et al. [19]. CLMs are similar to AAMs as both model shape and appearance variations of deformable objects like faces. CLMs model the appearance around each landmark in local templates, whereas AAMs model the whole object. The templates are normalized to have zero mean and unit variance.

When using the CLM to locate landmarks, the shape and appearance models are used to generate estimated locations and texture templates around each landmark. Then for each landmark the correlation between the template and the actual image at this position is computed. The iterative fitting process maximizes these correlations while respecting shape constraints. CLMs achieve a lower localization error than AAMs when predicting facial landmarks [19].

3. EXISTING SYSTEM

Heart related infections or Cardiovascular Diseases (CVDs) are the primary justification a colossal number of death on the planet in the course of the most recent couple of many years and has arisen as the most perilous illness, in India as well as in the entire world. Along these lines, there is a need of dependable, precise and attainable framework to analyze such sicknesses on schedule for legitimate therapy. AI calculations and methods have been applied to different clinical datasets to robotize the investigation of enormous and complex information. Numerous scientists, as of late, have been utilizing a few AI procedures to help the medical care industry and the experts in the analysis of heart related illnesses

4. PROPOSED SYSTEM

We smooth out AI calculations for powerful expectation of constant illness episode in disease-continuous networks. We try the adjusted expectation models over genuine medical clinic data collected from focal China in 2013–2015. To conquer the trouble of deficient information, we utilize a latent factor model to remake the missing information. We investigate a territorial persistent infection of cerebral infarction. We propose another convolutional neural organization (CNN)- based multimodal infection hazard prediction algorithm utilizing organized and unstructured information from medical clinic. As far as we could possibly know, none of the existing work zeroed in on both information types nearby clinical huge information investigation.

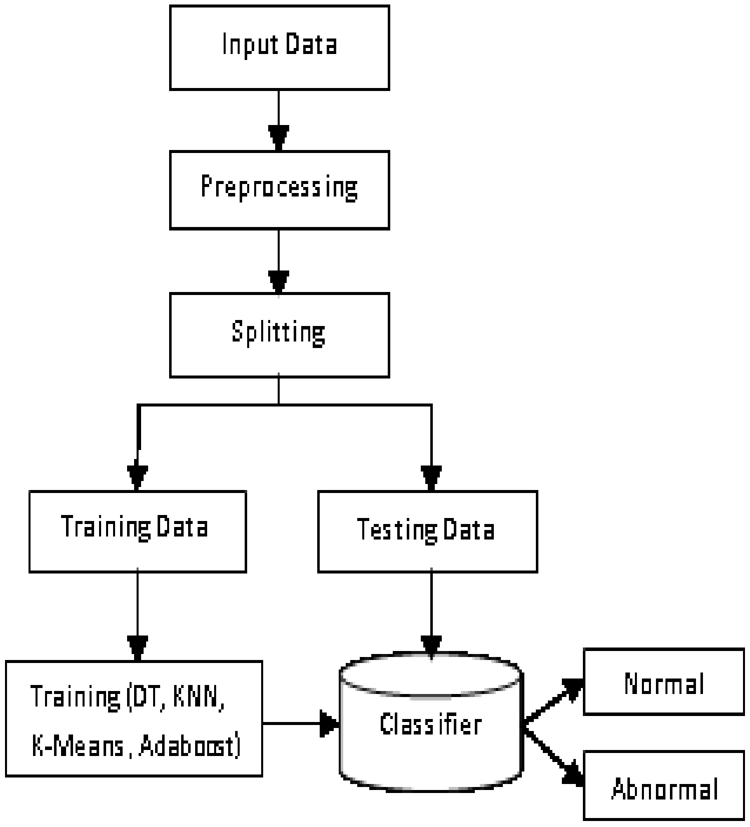


FIG 1 – SYSTEM ARCHITECTURE

5. METHODOLOGIES

In software development, a "module" typically refers to a self-contained unit of code that performs a specific function or encapsulates a set of related functionalities. Modules are an essential concept in modular programming, which aims to break down complex software systems into smaller, manageable, and reusable pieces. These smaller pieces, or modules, make it easier to develop, maintain, and understand the codebase. Modules can be implemented in various programming languages and are often organized hierarchically to form a well-structured software architecture.

Here are some key characteristics and principles related to modules in software development:

1. Encapsulation: Modules encapsulate related data and functions, allowing you to hide the internal implementation details and expose a clean, well-defined interface to other parts of the program.
2. Abstraction: Modules provide a level of abstraction, allowing you to work with high-level, understandable components without needing to understand the low-level implementation.
3. Reusability: Modular code is designed to be reusable. Modules can be used in different parts of the program or in other projects, reducing code duplication and improving maintainability.
4. Decomposition: A software system is decomposed into smaller, manageable modules, each responsible for a specific part of the system's functionality. This decomposition simplifies the development process.
5. Dependency Management: Modules can depend on each other, forming a dependency hierarchy. Managing dependencies helps ensure that changes in one module do not inadvertently affect others.
6. Isolation: Modules are isolated from each other, which means that changes made to one module do not directly impact other modules. This isolation enhances code stability and maintainability.
7. Testing: Modules can be tested individually, making it easier to identify and fix bugs. Isolating the code for testing simplifies the debugging process.
8. Documentation: Modules should be well-documented, describing their purpose, interfaces, and usage. This documentation helps other developers understand how to use the module correctly.
9. Standardization: Consistent naming conventions, coding styles, and design patterns are often applied to modules to ensure uniformity and readability across the codebase.
10. Scalability: Modular code is more scalable, as you can extend and modify the system by adding or replacing modules as needed, without affecting the entire codebase.

Examples of modules in software development can include:

- Utility Modules: Containing general-purpose functions or classes that can be used across the application, such as date manipulation, string handling, or mathematical operations.
- Database Modules: Handling database interactions, including connecting to databases, executing queries, and mapping database records to objects.
- User Interface Modules: Managing the graphical user interface (GUI) components, such as forms, buttons, and menus.
- Networking Modules: Handling communication with external servers or services over a network, such as making API requests.
- Security Modules: Implementing security features like authentication, authorization, and encryption.
- Logging and Error Handling Modules: Managing log files, capturing and reporting errors, and handling exceptions.
- Business Logic Modules: Implementing the core business logic of the application, including algorithms and

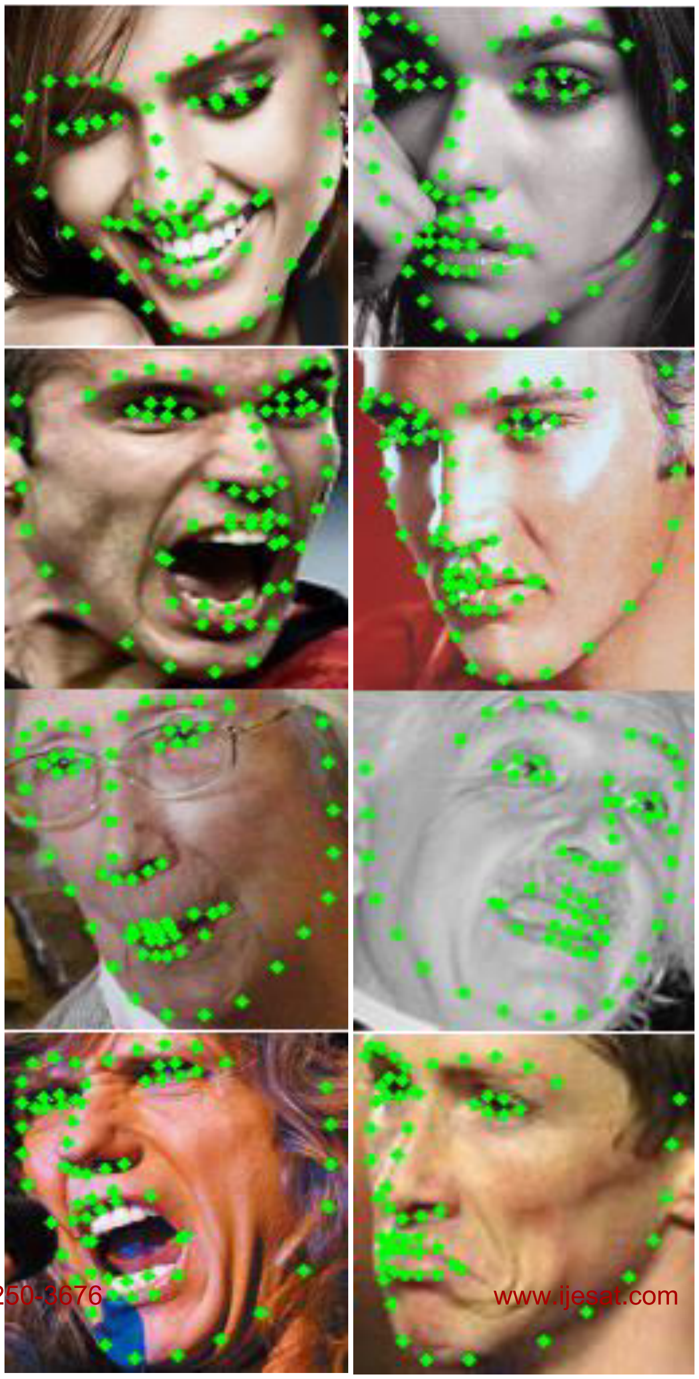
workflows specific to the domain.

In summary, modules are essential building blocks in software development that help organize, simplify, and maintain code. They promote code reusability, scalability, and easier testing while ensuring a structured and well-documented codebase.

6. RESULTS AND DISCUSSION SCREEN SHOTS

Screen shots

After the system has been evaluated using a lot of numbers, we finally show some landmarkpredictions that were created by the system developed in this work in Figure 2



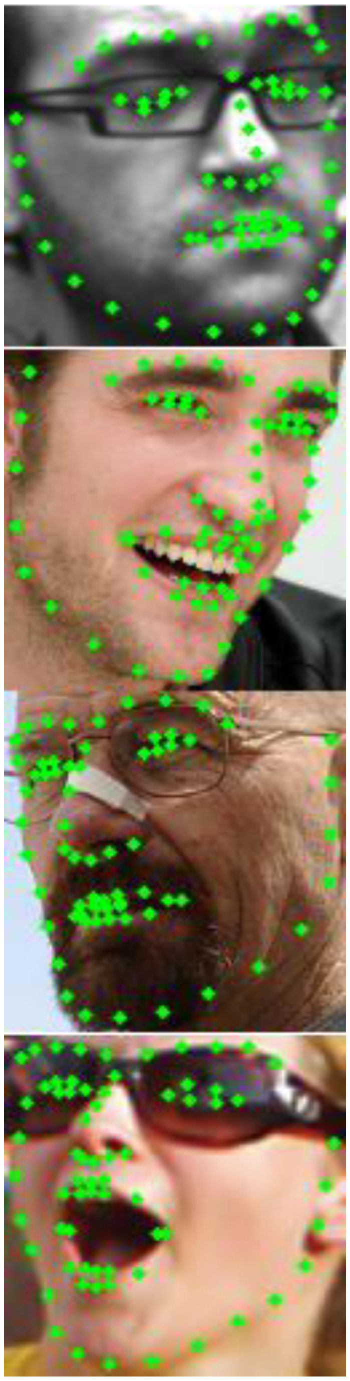


Fig.2: screen shots

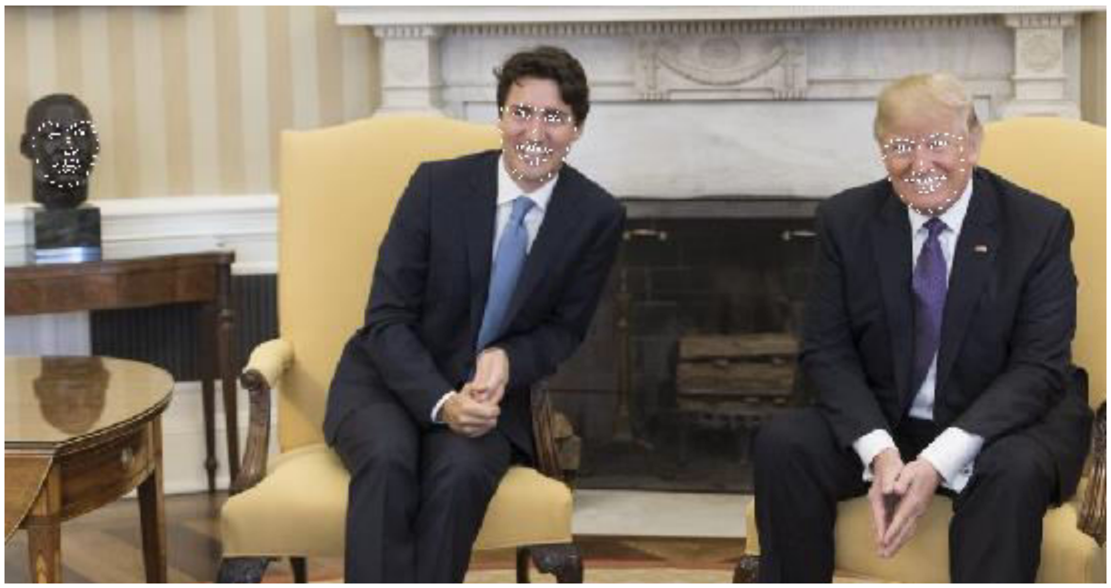


Fig.3 Facial Landmark Detection Sample

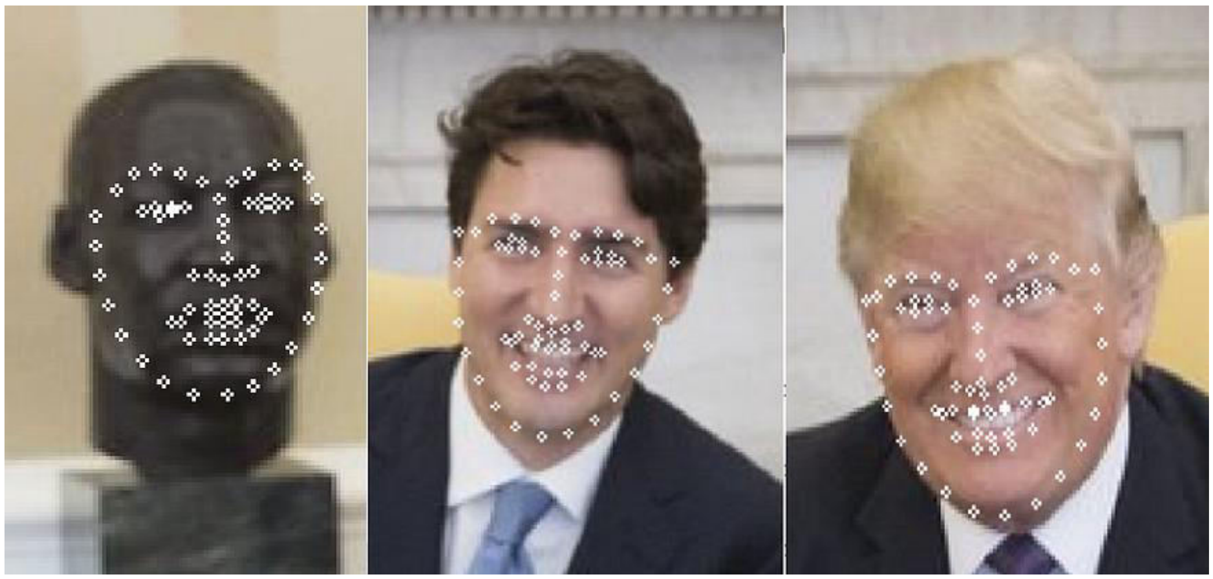


Fig.4 Zoomed output

7. CONCLUSION AND FUTURE SCOPE

CONCLUSION

This work aimed at improving state-of-the-art results for the task of facial landmark detection on unconstrained frontal and semi-frontal face images. To this end, a system consisting of two main parts has been designed, implemented and evaluated. As stacked Hourglass Networks (HGs) have already proven their effectiveness for other landmark pre- diction problems such as human pose estimation, we chose to use a stacked Hourglass (HG) to produce initial predictions for each facial landmark. In order to further refine these pre- dictions, we designed a Point Distribution Model (PDM).

We found that training the stacked HG using Wing loss and transforming the regressed heatmaps into numerical coordinates using a Differentiable Spatial To Numerical Trans- form (DSNT) is very effective for the task at hand. Various design choices for the stacked HG were evaluated. Regularizing the heatmaps using the Jensen-Shannon divergence proved to be helpful. Augmenting the training data by randomly rotating the face images lowered the prediction error especially for difficult images by a large factor. By applying the findings of the evaluation, our stacked HG is able to produce state-of-the-art results in terms of the point-to-point normalized error on both the 300-W and Menpo datasets. The error on Menpo was determined by a cross-dataset evaluation on a model that was only trained on 300-W.

The second part of this work focused on the PDM. We found that simple decoder networks with only two layers are able to model a mapping from a latent vector to the face shapes that can be used to improve the predictions made by the stacked HG. We furthermore have shown that initializing the shape parameters using a separate initializer network can lead to faster and more accurate predictions. Combining the stacked HG with the PDM further improves the prediction accuracy, especially for the inner facial landmarks on difficult images that show faces with strong head poses.

The stacked HG achieves state-of-the-art results on 300-W and Menpo both with and without the PDM. While adding the PDM does not lower the prediction error of the stacked HG for landmarks on the outline, it improves the localization accuracy on the inner facial landmarks of the difficult samples from 300-W. Our overall system is able to produce accurate predictions for faces in unconstrained settings, which includes faces with partial occlusions.

FUTURE SCOPE

The term "feature scope" refers to the extent or range of functionality that a specific feature or component of a software project is expected to encompass. It defines the boundaries and capabilities of a particular feature within a larger software application or system. Understanding and defining the feature scope is crucial in software development.

By clearly defining the feature scope, you can effectively manage expectations, reduce misunderstandings, and increase the likelihood of successful feature development within your software project.

8. REFERENCES

1. Timo Ahonen, Abdenour Hadid, and Matti Pietik¨ainen. “Face recognition with local binary patterns”. In: European conference on computer vision. Springer. 2004, pp. 469–481.

2. Akshay Asthana, Stefanos Zafeiriou, Shiyang Cheng, and Maja Pantic. “Robust dis- criminative response map fitting with constrained local models”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2013, pp. 3444– 3451.

3. Tadas Baltrusaitis, Peter Robinson, and Louis-Philippe Morency. “Constrained local neural fields for robust facial landmark detection in the wild”. In: Proceedings of the IEEE international conference on computer vision workshops. 2013, pp. 354–361.

4. Ankan Bansal, Anirudh Nanduri, Carlos D Castillo, Rajeev Ranjan, and Rama Chellappa. “Umdfaces: An annotated face dataset for training deep networks”. In: 2017 IEEE International Joint Conference on Biometrics (IJCB). IEEE. 2017, pp. 464– 473.

5. Peter N Belhumeur, David W Jacobs, David J Kriegman, and Neeraj Kumar. “Localizing parts of faces using a consensus of exemplars”. In: IEEE transactions on pattern analysis and machine intelligence 35.12 (2013), pp. 2930–2940.

6. Matteo Bodini. “A review of facial landmark extraction in 2d images and videos using deep learning”. In: Big Data and Cognitive Computing 3.1 (2019), p. 14.

7. Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. “A training algorithm for optimal margin classifiers”. In: Proceedings of the fifth annual workshop on Computational learning theory. ACM. 1992, pp. 144–152