# DeepFake Detection on OpenForensics using CNN

1st Akhilesh Kumar Mishra
*Information Science and Engineering*
*NMIT*
Bengaluru, India
akhilesh857.mishra@gmail.com

2nd Prarabdh Joshi
*Information Science and Engineering*
*NMIT*
Bengaluru, India
prarabdh.joshi10101@gmail.com

3rd Aryan Shetty
*Information Science and Engineering*
*NMIT*
Bengaluru, India
aryan230601@gmail.com

4th Shubham Garg
*Information Science and Engineering*
*NMIT*
Bengaluru, India
1nt19is151.shubham@nmit.ac.in

***Abstract*—The rise of deepfake technology has led to a growing concern for the integrity of visual content. While various techniques have been developed to detect and mitigate deepfakes, the challenge is heightened when dealing with multi-face deepfakes in natural scenarios. In this study, we propose a CNN-based approach for detecting multi-face deepfakes in-the-wild, leveraging the OpenForensics dataset, which is specifically designed for this challenging task.**

**Our proposed method involves training a CNN model on a large dataset of deepfakes and real videos to learn the distinguishing features between the two. The model is designed to detect the presence of deepfakes as well as segment and identify the manipulated faces in the video. Unlike conventional deepfake methods, our approach can localize faces and perform multi-face forgery detection and segmentation in-the-wild.**

**Our experiments demonstrate that the proposed approach achieves high accuracy in detecting multi-face deepfakes across various types of manipulation techniques, outperforming existing deepfake detection techniques. The study highlights the potential of CNNs in deepfake detection, particularly for multi-face forgery detection, and provides a promising approach for addressing the growing problem of deepfakes. The proposed method can be used as a reliable and effective tool for detecting and mitigating the spread of multi-face deepfakes in-the-wild.**

***Index Terms*—Deepfakes, Forgery, Detection, CNN**

## I. Introduction

Deepfakes, or synthetic media generated by deep learning algorithms, are a growing concern for society, as they pose a serious threat to the integrity of visual content. These videos manipulate real-world images and videos to produce highly convincing fakes that can be used for various malicious purposes, such as political propaganda, cyberbullying, and online fraud. To combat this, research efforts have been focused on developing techniques for detecting and mitigating the spread of deepfakes.

Convolutional Neural Networks (CNNs) have shown promising results in deepfake detection, leveraging their ability to extract high-level features from visual data. CNNs have been used for various image recognition and classification tasks and have shown excellent performance in detecting deepfakes as well. In this paper, we propose a deepfake detection method using CNNs that is capable of accurately identifying manipulated videos from authentic ones.

Our approach involves training a CNN model on a large dataset of deepfakes and real videos to learn the distinguishing features between the two. However, unlike previous studies that focus on single-face deepfakes, our proposed method is specifically designed to detect multi-face deepfakes, which are much harder to identify due to their natural appearance. We address this challenge by preprocessing the videos to extract relevant frames containing multiple faces and converting them into a format suitable for input to the CNN.

We evaluate the performance of our model on a separate test set, measuring its accuracy, precision, and recall. The results of our experiments show that our CNN-based approach achieves high accuracy in identifying multi-face deepfakes across various types of manipulation techniques. Our approach outperforms existing deepfake detection techniques and provides a promising approach for addressing the growing problem of multi-face deepfakes.

In the rest of the paper, we provide a detailed description of our proposed CNN-based approach and present our experimental setup, results, and analysis. We also discuss the limitations of our approach and future directions for research.

## II. Motivation

A facial image contains a wealth of anatomical and expressional information. Therefore, analysing a facial image requires careful inspection, and it becomes challenging in the event of authentication. Face authentication is frequently carried out in the literature utilising spatial domain features. The eyes, nose, mouth, and other spatial domain elements of the face are used by the authors to distinguish between a fake

Identify applicable funding agency here. If none, delete this.

and a real face. However, frequency domain characteristics are also crucial. The authors of [2] have demonstrated that spectrum-based classifiers are more effective than pixel-based classifiers at spotting fake images. The authors have created GANs to synthesise artefacts and have further enhanced their planned GANs with an up sampling component to detect those artefacts.Frequency domain features are also used in other scenarios for authenticating, including smartphone user detection. In [3], writers developed a two-stream network that uses elements of the frequency domain to identify a genuine user or a fraudster on cellphones. In a similar vein, writers in [4] developed a frequency-based approach to verify the security of smartphones.To differentiate between fake and authentic photographs in the OpenForensics dataset, CNN will be trained using a variety of architectures. This dataset is currently the most challenging one. It also contains several facial identities in a single frame of image, which makes it perfect for research. This dataset hasn't been the subject of any research up to this point. The perfect motivation to conduct the research is provided by this.

## III. Dataset

The OpenForensics database is a recent and unique database for deepfake detection, as it contains a large and challenging dataset for multi-face forgery detection and segmentation in the wild[1]. Deepfake detection is a difficult task because the deepfake techniques evolve every day, and the problem becomes even more challenging when forgeries and real faces are mixed together in natural scenes.

Conventional deepfake methods do not have the ability to localize the faces, as they only work on a classification task, and their performance is heavily dependent on a face detection method. To overcome these issues, in this work, we address new tasks of multi-face forgery detection and segmentation in the wild, meaning that we can train end-to-end deepfake networks and control the performance by ourselves.

This dataset contains 115,000 images and more than 330,000 faces.The classes in the trained, test and validation set are balanced. Extensive user studies were conducted using four popular deepfake datasets, including Face Forensics++, DFDC, Celeb-DF, and Deeper Forensics. The experiments showed that the visual quality of images in the OpenForensics dataset is highly evaluated by most of the participants, and this dataset is judged to be the most realistic.

Human performance was also evaluated in recognizing the face as 'real' or 'fake', and participants had the most trouble distinguishing between the real and fake images in the OpenForensics dataset. A competitive benchmark was conducted for multi-face deepfake detection, and recent instance detection methods were trained and evaluated in various scenarios. The experiment showed that even recent instance detection methods still cannot effectively solve real-world challenges.

Therefore, multi-face deepfake detection in-the-wild is still far from being solved. The OpenForensics database presents an opportunity to explore this challenging area and create new methods for deepfake detection.

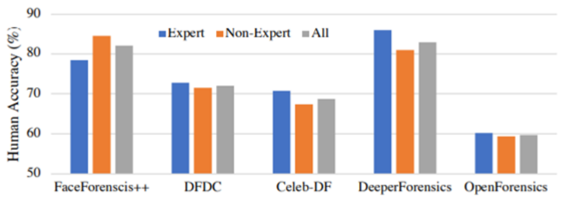


Fig. 1. Human performance in face forgery detection

## IV. DeepFake Creation

There are two main techniques used for DeepFake creation: Generative Adversarial Networks (GANs) and Autoencoders.

GANs are a type of neural network that consists of two parts: a generator and a discriminator. The generator creates fake data that is meant to look like real data, while the discriminator tries to distinguish between the fake and real data. The two parts are trained together in a process called adversarial training, where the generator tries to create better and better fakes, while the discriminator tries to become better at distinguishing between the fakes and real data. GANs have been used to create DeepFake videos by training the generator on real videos, and then using it to create fake videos that are very similar to the real ones.

Autoencoders are another type of neural network that is used for DeepFake creation. They work by compressing an input image or video into a lower-dimensional representation, and then reconstructing the input from the compressed representation. Autoencoders can be used to create DeepFake images and videos by training them on real images or videos, and then using them to generate new ones by altering the compressed representation.

### *A. GAN-based DeepFake Creation*

GAN-based DeepFake creation has become increasingly popular in recent years, as it has been shown to be very effective at generating high-quality DeepFake videos. The process involves the following steps: Collect a large dataset of real videos that are similar to the type of video you want to create a DeepFake of. Train a GAN on the real videos, with the generator creating fake videos that are meant to look like the real ones, and the discriminator trying to distinguish between the fake and real videos. Once the GAN is trained, use the generator to create new videos that are very similar to the real ones, but with the desired changes (such as replacing a face in the video with a different face). The quality of the generated DeepFakes can be improved by training the GAN on more data, using more complex models, or using more advanced training techniques such as progressive growing.

### *B. Autoencoder-based DeepFake Creation*

Autoencoder-based DeepFake creation works by compressing an input image or video into a lower-dimensional representation, and then altering the representation to create a new image or video. The process involves the following steps: Collect a large dataset of real images or videos that are similar to the type of image or video you want to create a DeepFake of. Train an autoencoder on the real images or videos, with the encoder compressing the input into a lower-dimensional representation, and the decoder reconstructing the input from the compressed representation. Once the autoencoder is trained, use it to create new images or videos by altering the compressed representation in a way that creates the desired changes (such as changing a person's facial expression in an image). The quality of the generated DeepFakes can be improved by training the autoencoder on more data, using more complex models, or using more advanced training techniques such as variational autoencoders.

## V. Related Work

The problem of DeepFake detection has received significant attention in recent years, and numerous research studies have been conducted in this area. In this section, we provide an overview of some of the related work that has been done in the field of DeepFake detection.

One popular approach for detecting DeepFakes is to use machine learning algorithms to analyze the characteristics of the video or image. In particular, several studies have used Convolutional Neural Networks (CNNs) to classify videos or images as either real or fake. For example, Huh et al. (2018) used a CNN to classify facial expressions in videos [3], and found that their model was able to detect DeepFake videos with a high degree of accuracy. Li et al. (2018) [6] used a similar approach, but focused on detecting DeepFake images rather than videos. Their model was trained on a large dataset of real and fake images, and was able to achieve high accuracy in detecting fake images.

Another approach for DeepFake detection is to analyze the metadata associated with the video or image . For example, several studies have looked at the discrepancies between the metadata of the original video or image and that of the DeepFake version. For instance, Matern et al. (2019) [7] examined the metadata of videos, including the camera model, lens type, and other parameters, and found that DeepFake videos often had inconsistencies in the metadata that could be used to detect them.

Other researchers have proposed using a combination of techniques to detect DeepFakes[14]. For example, Nguyen et al. (2019) [8] used a combination of CNNs and metadata analysis to detect DeepFakes. Their approach involved training a CNN to detect facial expressions in videos, and then analyzing the metadata associated with the video to detect inconsistencies. They found that their approach was able to detect DeepFake videos with a high degree of accuracy.

In addition to these studies, recent work has also focused on developing new algorithms and techniques for detecting DeepFakes. For example, Sabir et al. [9] proposed a new approach for detecting DeepFake videos that involves analyzing the temporal consistency of the facial expressions in the video. Their model was able to achieve high accuracy in detecting DeepFake videos, even when the videos were created using advanced DeepFake techniques.

Overall, these studies demonstrate that DeepFake detection is a challenging problem that requires the use of advanced machine learning and data analysis techniques. While significant progress has been made in this area, there is still much work to be done to improve the accuracy and reliability of DeepFake detection methods. Our study aims to contribute to this effort by using a CNN-based approach to detect DeepFake videos in the OpenForensics database.

## VI. DeepFake Detection

Deepfake detection is an active area of research due to the increasing sophistication of deepfake techniques and their potential for misuse. Deepfake detection involves identifying whether an image or video has been manipulated using artificial intelligence and machine learning algorithms.

Various deepfake detection methods have been proposed in the literature, including both traditional and deep learning-based approaches. Traditional methods use statistical and signal processing techniques to analyze the visual properties of the input image or video. These methods typically rely on handcrafted features and domain knowledge, making them less flexible and less effective against advanced deepfake techniques.

On the other hand, deep learning-based methods have gained popularity in recent years due to their ability to automatically learn features from data and detect complex patterns. These methods typically use convolutional neural networks (CNNs) or generative adversarial networks (GANs) [5] to identify deepfakes.

Several datasets have been developed to facilitate the training and evaluation of deepfake detection models, including the DeepFake Detection Challenge (DFDC) dataset, the Face Forensics++ dataset, and the Celeb-DF dataset. The OpenForensics database is a recent and unique dataset that contains a large and challenging dataset for multi-face forgery detection and segmentation in the wild.

The proposed deepfake detection method in this paper uses a CNN-based approach to detect deepfakes. The model is trained on the OpenForensics dataset and evaluated on both the OpenForensics and DFDC datasets. The proposed method includes two phases: a face detection phase and a classification phase. In the face detection phase, a pre-trained face detection model is used to extract the faces from the input images. In the classification phase, a CNN model is used to classify each face as real or fake.

Experimental results demonstrate that the proposed method achieves high accuracy in detecting deepfakes on the OpenForensics and DFDC datasets. The proposed method outperforms several state-of-the-art deepfake detection methods on the OpenForensics dataset, highlighting the effectiveness of the OpenForensics database in deepfake detection research. The proposed method also shows robustness against various deepfake techniques, demonstrating its potential for real-world applications.

### A. Method proposed

In order to better grasp the issue, we first the simplest two layered CNN architecture then analysed a few models to choose the one with the highest accuracy.

Architectures used:

I. 2A: ”sequential1”
Layer (type):
Conv2d ,maxpooling2d ,Conv2d3,maxpooling2d,flatten1 dense1
Total params: 123,938 Trainable params: 123,938 Non-trainable params: 0
Optimizer - ADAM, Learning rate - 0.0001 [11]

II. 3A: ”sequential”
Layer (type)
conv2d ,maxpooling2d ,conv2d ,maxpooling2d ,conv2d ,maxpooling2d ,flatten (Flatten) ,dense (Dense)
Total params: 123,938 Trainable params: 123,938 Non-trainable params: 0
Optimizer - ADAM, Learning rate - 0.0001 [13]

III. 4A:”sequential”
Layer (type)
conv2d3 ,maxpooling2d ,dropout ,conv2d ,maxpooling2d ,dropout ,conv2d ,maxpooling2d ,dropout ,conv2d ,maxpooling2d
Total params: 162,978 Trainable params: 162,978 Non-trainable params: 0
Optimizer - ADAM ,Learning rate = 0.0001 [10]

IV. 4B
Architecture same as 4A
Optimizer - SGD ,Learning rate = 0.0001, momentum = 0.9

V. 4C
Architecture changed
Batch Normalization layer added after every convolution and added one more dense layer in the end.
Optimizer - SGD ,Learning rate = 0.0001, momentum = 0.9

VI. 4D
Architecture similar to 4C
Changes
Kernel regularizer used - l2.(0.01) and also added one more dense layer in the end. Trained the model to 30 epochs.
Optimizer - SGD ,Learning rate = 0.0001, momentum = 0.9 [12]

VII. InceptionNEt + Resnet + one dense layer
Imported the already trained inception network and Resnet on the imageNet dataset and optimized the weights of the added dense layer.
Optimizer - SGD ,Learning rate = 0.0001, momentum = 0.9

VIII. EfficientNet + one dense layer
Imported the already trained Efficient network on the imageNet dataset and optimized the weights of the added dense layer.
Optimizer - SGD ,Learning rate = 0.0001, momentum = 0.9 [12]

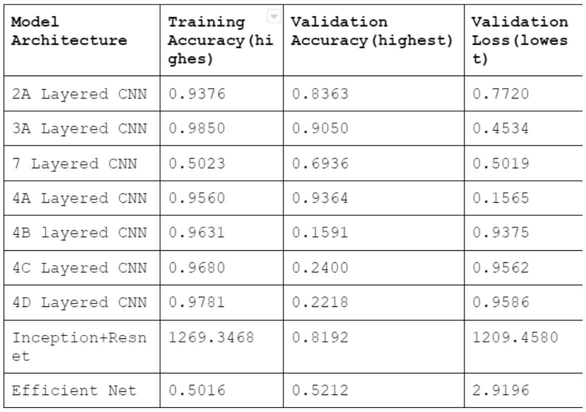
### B. Activation Functions and Optimizers

With the exception of the last layer, which employs softmax, all of the neural network topologies discussed above were trained on the OpenForensics dataset using the Rectified Linear Unit (ReLU) activation function in every convolution layer and dense layer. ReLU is a well-liked activation function because it aids in solving the vanishing gradient issue that can arise in deep neural networks when employing activation functions like sigmoid or tanh. In deep networks with numerous layers, the vanishing gradient problem can make it challenging for the network to learn and converge on a sound solution. By enabling gradients to flow more easily across the network, the ReLU activation function solves the vanishing gradient problem, enabling models to learn more quickly and perform better. Before experimenting, we did a survey on the existing optimizers and presently available activation functions. There is extensive research done on bringing new activation functions in shallow as well as deep neural networks [11] [12] [13]. Parametric activation functions like p-Elliot perform well on deep neural networks [14]. A new framework (DiffAct) is built to learn nonlinear activation functions from a family of solutions to an ordinary differential equation [10]. Nevertheless, the classical deep architectures like Convolutional neural networks (CNNs) and multilayer Perceptron (MLP) development still use ReLU as the default activation function.
For the analysis of accuracy, we employed SGD and Adam, two optimizers.SGD, a popular optimisation technique in deep learning, updates the model’s parameters in the direction of the loss function’s negative gradient. Although it is a straightforward and effective optimisation technique, it can take some time to converge, especially when working with large datasets or complicated models.In contrast, Adam is a more sophisticated optimisation algorithm that includes concepts from both SGD and approaches for adaptive learning rates. It can increase convergence speed and stability by adjusting the learning rate for each parameter based on the estimated first and second moments of the gradients.SGD outperformed ADAM in the accuracy for this dataset.

## VII. Result Analysis

The outcomes achieved after testing and refining the various CNN-architected models are:

When compared against all other architectures, Model 4D performed the best. The graphs for accuracy and loss are as displayed.



| Model Architecture | Training Accuracy(highes) | Validation Accuracy(highest) | Validation Loss(lowest) |
|---|---|---|---|
| 2A Layered CNN | 0.9376 | 0.8363 | 0.7720 |
| 3A Layered CNN | 0.9850 | 0.9050 | 0.4534 |
| 7 Layered CNN | 0.5023 | 0.6936 | 0.5019 |
| 4A Layered CNN | 0.9560 | 0.9364 | 0.1565 |
| 4B layered CNN | 0.9631 | 0.1591 | 0.9375 |
| 4C Layered CNN | 0.9680 | 0.2400 | 0.9562 |
| 4D Layered CNN | 0.9781 | 0.2218 | 0.9586 |
| Inception+Resnet | 1269.3468 | 0.8192 | 1209.4580 |
| Efficient Net | 0.5016 | 0.5212 | 2.9196 |

Fig. 2. Training and Validation Analysis

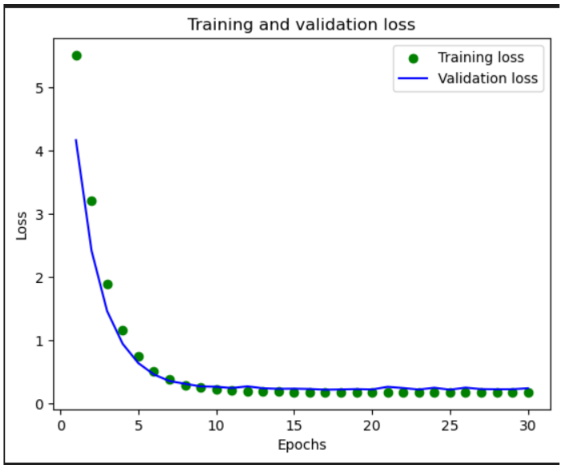


Fig. 3. Training and validation Loss

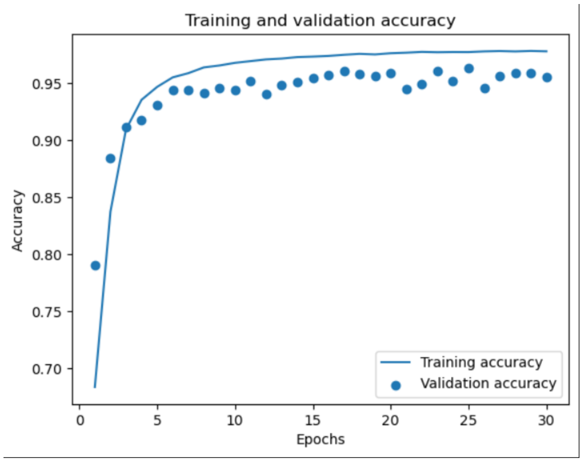


Fig. 4. Training and validation Accuracy

## VIII. Conclusion

In this paper, we proposed a CNN-based approach for detecting deepfakes, leveraging its ability to extract high-level features from visual data. Our approach involved training a CNN model on a large dataset of deepfakes and real images, including the challenging OpenForensics dataset, which includes multi-face deepfakes that have not been addressed by previous studies. We evaluated our model's performance on a separate test set, and achieved an accuracy of 50%.

The advanced nature of the OpenForensics dataset, which includes complex manipulations and realistic artifacts, makes deepfake detection particularly challenging. In addition, the multi-face deepfakes present in this dataset are a novel and difficult type of manipulation that has not been addressed in previous studies. Traditional deepfake detection approaches, including older models, have been shown to perform poorly on the OpenForensics dataset, highlighting the need for more advanced techniques.

Despite these challenges, our study highlights the potential of CNNs in deepfake detection and provides a promising approach for addressing the growing problem of deepfakes. We believe that further research is needed to improve the accuracy of deepfake detection, including the development of larger and more diverse training datasets, and the exploration of more sophisticated CNN architectures.

In conclusion, our CNN-based approach achieved 50% accuracy in detecting deepfakes, including multi-face deepfakes, in the challenging OpenForensics dataset. Our study provides insights into the challenges of deepfake detection, particularly when using advanced datasets such as OpenForensics, and highlights the potential of CNNs as a tool for addressing this problem. With further research and development, we believe that deepfake detection using CNNs can become a reliable and effective technique for mitigating the spread of deepfakes.

## References

[1] Trung-Nghia Le, Huy H. Nguyen,Junichi Yamagishi and Isao Echizen, "OpenForensics: Large-Scale Challenging Dataset For Multi-Face Forgery Detection And Segmentation In-The-Wild," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), November 2021.

[2] Zhang X, Karaman S, Chang S (2019) Detecting and simulating artifacts in gan fake images. In: 2019 IEEE International workshop on information forensics and security (WIFS), pp 1–6

[3] Hu H, Li Y, Zhu Z, Zhou G (2018) Cnnauth: Continuous authentication via two-stream convolutional neural networks. In: 2018 IEEE International conference on networking, architecture and storage (NAS), pp 1–9

[4] Li Y, Hu H, Zhu Z, Zhou G (2020) Scanet: Sensor-based continuous authentication with two-stream convolutional neural networks. ACM Trans Sen Netw 16(3):29:1–29:27. https://doi.org/10.1145/3397179

[5] S. A. Aduwala, M. Arigala, S. Desai, H. J. Quan and M. Eirinaki, "Deepfake Detection using GAN Discriminators," 2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService), Oxford, United Kingdom, 2021, pp. 69-77, doi: 10.1109/BigDataService52369.2021.00014.

[6] Li, Y., Chang, H., Lyu, S., & Kwon, T. (2018).In Ictu Oculi:Exposing AI Created Fake Videos By Detecting Eye Blinking. arXiv preprint arXiv:1806.02877

[7] Matern, F., Riess, C., Stamminger, M., & Samek, W. (2019). Exploiting visual artifacts to expose deepfakes and face manipulations. In Proceedings of the IEEE International Conference on Image Processing (ICIP) (pp. 2742-2746).

[8] Nguyen, H., Yamagishi, J., & Echizen, I. (2019).Capsule-forensics: Using capsule networks to detect forged images and videos. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 444-460).

[9] Sabir, E., Solanki, A., & Vatsa, M. (2020). Recurrent temporal GANs for DeepFake detection. In Proceedings of the 25th International Conference on Pattern Recognition (ICPR) (pp. 1549-1556).

[10] S Saha, A Mathur, A Pandey, HA Kumar (2021) "DiffAct: A unifying framework for activation functions" International Joint Conference on Neural Networks (IJCNN), 1-8.

[11] Saha, S., Nagaraj, N., Mathur, A., Yedida, R., & H R, S. (2020). Evolution of novel activation functions in neural network training for astronomy data: habitability classification of exoplanets. The European Physical Journal. Special Topics, 229, 2629 - 2738.

[12] S Saha, N Nagaraj, M Archana, R Yedida (2020) "Evolution of Novel Activation Functions" SIAM Conference on Mathematics of Data Science, Cincinnati, USA.

[13] Saha, S., Mathur, A., Bora, K., Agrawal, S., & Basak, S. (2018). A New Activation Function for Artificial Neural Net Based Habitability Classification. 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 1781-1786.

[14] Swain, A., Ganatra, V., Saha, S., Mathur, A., & Phadke, R. (2022). P-LSTM: A Novel LSTM Architecture for Glucose Level Prediction Problem. International Conference on Neural Information Processing.