MobiDeep: Mobile DeepFake Detection through Machine Learning-based Corneal-Specular Backscattering

Muhammad Mohzary[1,2], Khalid J Almalki[3], Baek-Young Choi[1], Sejun Song[1]
[1]School of Science and Engineering, University of Missouri-Kansas City, MO, USA
[2]Department of Computer Science, Jazan University, Jazan, Saudi Arabia
[3]College of Computing and Informatics, Saudi Electronic University, Riyadh, Saudi Arabia

***Abstract*—DeepFake has accomplished notable advancement with the AI-leveraged production and manipulation techniques of fictitious human facial images. Despite many benign and fun applications, the generated fake images can negatively influence the authenticity of online information by originating deception, manipulation, persecution, and seduction, defying societal quality and human rights, which becomes critical security and privacy threat in social networks. Hence, real-time DeepFake detection and limitation technologies on the mobile platform are essential to building a controlled, harmless DeepFake ecosystem. This paper presents a real-time, cloudless, lightweight mobile app for human visual DeepFake detection using machine learning technologies named MobiDeep (Mobile DeepFake Detection through Machine Learning-based Corneal-Specular Backscattering). MobiDeep stems from a hypothesis that the existing DeepFake creation methods, including replacement, editing, and synthesis, lack the ensemble with the reflective objects. Focusing on the most reflective area of a human face, corneal-specular backscatter images of eyes, we seek the similarity and consistency with multiple surrounding environment features, including color components, shapes, and textures. We have implemented a cross-platform mobile application to evaluate the performance using various input parameters and lightweight Deep Neural Network (DNN) architectures. The empirical results show that MobiDeep achieves high accuracy (98.7%) and rapid detection speed (less than 200 ms) in detecting sophisticated DeepFake images within a subsecond.**

***Index Terms*—DeepFake, Corneal-Specular Backscattering.**

I. Introduction

DeepFake techniques generating AI-leveraged fictitious human facial images, have garnered increasing attention for their diverse usage scenarios. Although many benign applications such as funny jokes and visual humans, **DeepFake** can be malignantly used by spying on people with fake identities over social media, creating humiliating and nonconsensual fake images, spreading fake news, and planning scams and financial fraud, which becomes **serious security and privacy threat in social networks**. As DeepFake generation technologies have been improving sophisticatedly, it is getting difficult to differentiate the falsified images by bare human eyes. Furthermore, due to the recent advancement of the mobile DeepFake applications such as Reface [1], Avatarify [2], and Wombo [3], making realistic DeepFake images and videos has become astonishingly easy. Tens of millions of clips are generated every day on social networks. DeepFakes are ready

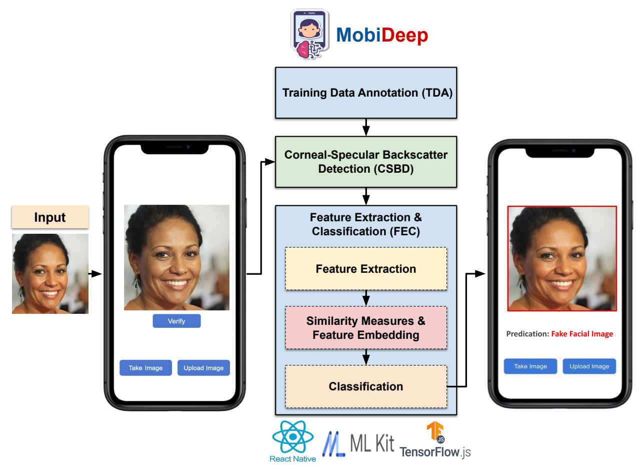

Fig. 1. MobiDeep DeepFake Detection Method.

to disrupt and diminish authenticity, privacy, and security for our society and worsen when the Internet becomes an immersive metaverse. The current DeepFake detection methods [4]–[7] lack the transferability to unseen cases and become overfitted to low-quality datasets due to the limited training on low-quality videos with easy-to-detect artifacts such as shapes or visible boundaries of the fakes. Similarly, eye-based DeepFake detection methods [8]–[10] cannot generalize well when confronting sophisticated DeepFake media because they only consider single artifacts of eyes, either iris color, blinks, or similarity of corneal reflections on both eyes. KaiCatch [11] is the most recent DeepFake detection mobile application. However, it is a cloud-based service that takes a few days to get a classification result. Hence, real-time DeepFake detection and limitation technologies are essential to prevent the imaginable chaos that manipulates the incapacity to discern DeepFake images and videos.

This paper presents a real-time, cloudless, lightweight mobile DeepFake detection technology named **MobiDeep** (**Mobi**le **Deep**Fake Detection through Machine Learning-based Corneal-Specular Backscattering), shown in Figure 1. We hypothesized that the existing DeepFake generation techniques, including replacement, editing, and synthesis, are hard to coordinate their counterfeits with the reflective elements, as presented in Figure 2 (a) and (d). Therefore, we focus on the most reflective area of a human face, corneal-specular backscatter images of eyes. We seek the similarity and consistency of corneal-specular backscatters with multiple surrounding semantics, such as illumination and environmental conditions that are hard to fake. Thus, we extract numerous features, including corneal-specular backscatter images' color components, shapes, and textures, instead of checking a single aspect of the eyes, such as the similarity of corneal reflections on both eyes. Furthermore, we extract Facial Image Environmental Parameters (FIEP) to check the ensemble of the reflectance with the surrounding environmental factors such as indoor/outdoor, bright/dark, backgrounds, and strength and direction of light. MobiDeep embeds the *FIEP* into the feature extraction and classification process to detect the symmetricity and consistency in both eyes' color components and reflection patterns. As illustrated in Figure 1, we have implemented a cross-platform mobile application to evaluate the performance using various input parameters and lightweight Deep Neural Network (DNN) architectures. MobiDeep consists of a couple of ML components, including Training Data Annotation (TDA), Corneal-Specular Backscatter Detection (CSBD), and Feature Extraction and Classification (FEC). CSBD detects a face area and surrounding scenes from the input images to identify *CSB* images and extracts *FIEP* features. FEC extracts corneal highlight features from the *CSB* images, measures the *CSB*s symmetry and color consistency, embeds additional *FIEP* features, and classifies the *CSB* images as fake or real. We use Siamese Convolutional Neural Networks (SCNN) with three most lightweight CNN backbones including MobileNet-V2 [12], EfficientNet-B0 [13], and DenseNet-121 [14] for the feature extraction. We also create a new MobiDeep DeepFake Detection (MobiDeep-DFD) dataset, including real and fake images to annotate it with various *CSB* information for corneal highlights segmentation. The experimental results using MobileNet-V2 with the MobiDeep-DFD dataset show that MobiDeep achieved a high accuracy (98.70%) and fast classification speed (less than 200 ms) in detecting sophisticated DeepFake images on various mobile devices.

The main contributions of this work include the following:

- A lightweight real-time mobile application is designed to cope with the cloudless ML approach by modularizing feature extraction and embedding.
- High accuracy and fast classification speed DeepFake detection application is implemented in the mobile environment.
- A high-quality DeepFake Detection dataset is collected and annotated for corneal highlight segmentation and DeepFake detection applications.
- ML methods are proposed to build an ensemble with multiple surrounding reflective features, and the impact of environmental factors on reflectance is evaluated.

The remainder of this paper is organized as follows. Section II describes the existing DeepFake detection methods.

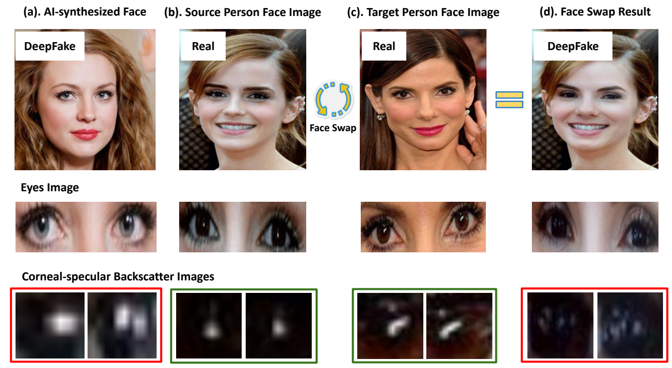

Fig. 2. Samples of Real and DeepFake Facial Images with their Reflective Elements (the Corneal-Specular Backscatter Images of Eyes): (a) is AI-synthesized Face From [15], (b) and (c) are both Real, (d) is a DeepFake Face Generated Using the Face Swapper Online Tool [16], Face Swapper Replaces the Target person's (c) Facial Landmarks with that of a Source Person (b), in the Same Time it Preserves the Source Person's (b) Identity.

Section III explains the design of MobiDeep. Section IV discusses the experiment setups and results. Section V concludes the paper.

II. Related Work

This section presents the current DeepFake detection methods for mobile devices and discusses the current eye-based DeepFake detection techniques and their drawbacks.

A. DeepFake Detection on Mobile Devices

The Korea Advanced Institute of Science and Technology (KAIST) recently proposed a cloud-based mobile application service called KaiCatch [11] for DeepFake detection. However, KaiCatch requires users to download the KaiCatch mobile application and register the service to upload the images or videos to be tasted. After three to four days, a classification result (fake or real) will be sent back to the user, and for more detailed results sent via email, it charges $1.76 per image. Nowadays, anyone can make realistic DeepFake media using easy-to-use DeepFake creation mobile applications such as Reface, Avatarify, or Wombo. Using generated DeepFake media for defamation, blackmailing, and harming innocent individuals' credibility, necessitates having a real-time DeepFake detection mobile application to quickly and precisely identify forged media.

B. Eye-based DeepFake Detection Methods

Several DeepFake detection methods have focused on analyzing the eyes' visual features. For example, the authors in [10] used commonly available computer vision methods to identify GAN-synthesised faces by noticing that they may have inconsistent iris colors and the specular reflection from the eyes is either missing or appear as a white blob. However, such inconsistencies and manipulation artifacts have mainly been improved in the recent DeepFake generation models. Hu et al. [8] proposed a physiological and physical detection method that uses the inconsistency of the corneal-specular highlights

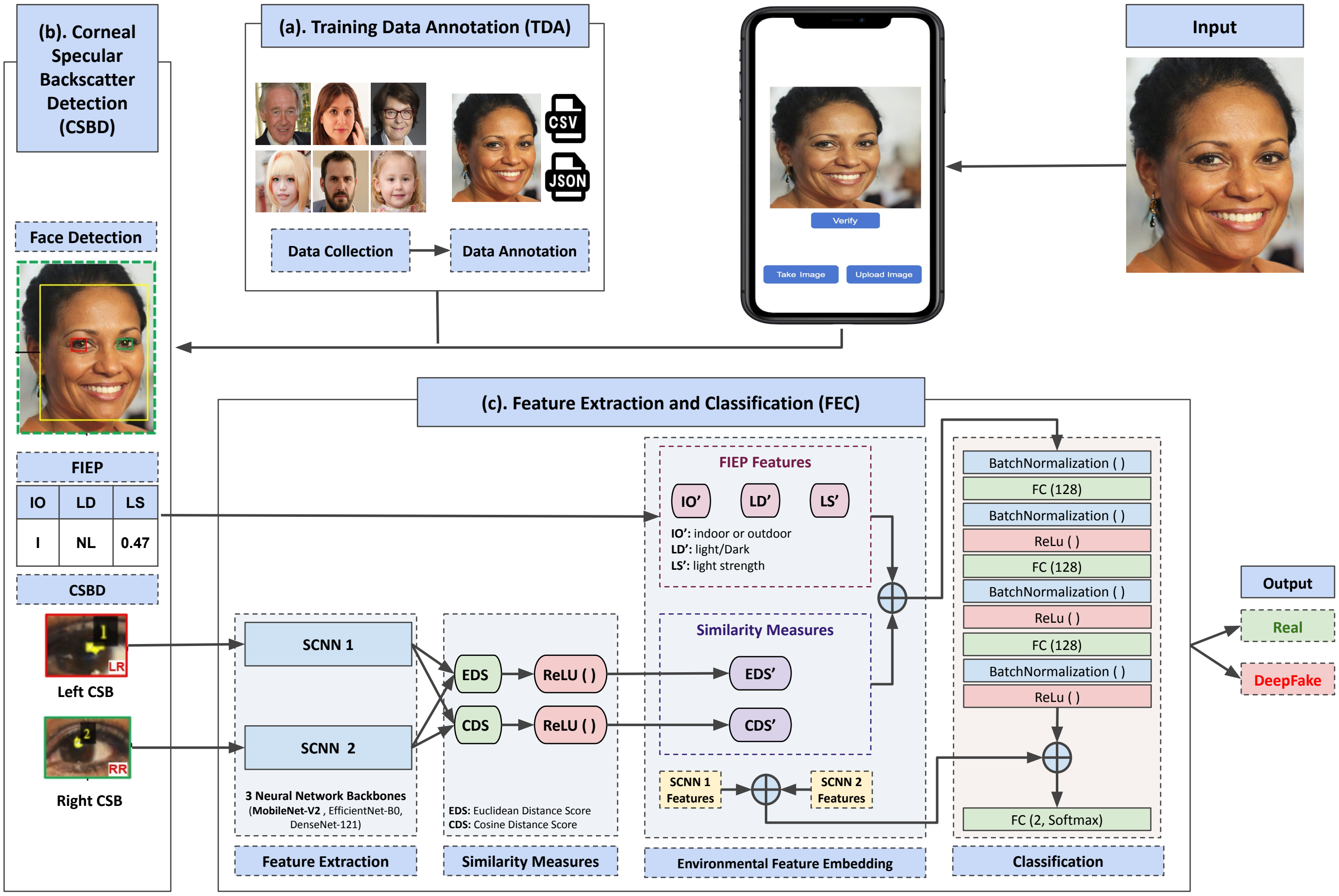

Fig. 3. The Block-diagram of MobiDeep DeepFake Detection Method.

between the two synthesized eyes. They assumed that two eyes see the same scene, and their corresponding corneal-specular highlights should exhibit strong similarities. Their experiments show a clear separation between the distribution of the similarity scores of the real and GAN-synthesized faces when strict portrait settings are followed, such as the two eyes have a frontal pose, the eyes are distant from the light or reflection source, and all light sources and reflectors are visible to both eyes. However, when the portrait setting is ignored, [8] will raise many false positives. In addition, their single factor (shape) similarity measures alone cannot be a strong indicator for classifying fake or real images. Hui et al. [5] also proposed a physiological detection method based on the irregular pupil shapes as a cue to distinguish between real and GAN-generated faces. However, this method will lead to wrong predictions when the shapes are non-elliptical in the real faces or there are occlusions on the pupil. Eyes-based DeepFake detection methods only focus on single artifacts of the eyes' visual features. Hence, they fail to detect sophisticated DeepFake reliably.

MobiDeep is designed to detect DeepFake efficiently on mobile devices using a lightweight machine learning model. It also coordinates various features (e.g., colors, edge, textures, etc.) of corneal-specular backscatter images. It embeds surrounding environmental factors, such as indoor/outdoor, bright/dark, and light's strength, and checks the ensemble with the reflectance.

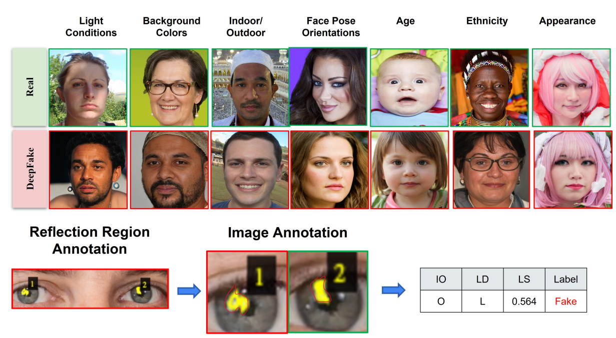

Fig. 4. Image Classification and Annotation.

III. MobiDeep Architecture

The principal objective of MobiDeep is to detect DeepFake by analyzing *CSB* images with multiple surrounding environmental parameters. MobiDeep mainly consists of Training

Data Annotation (TDA), Corneal-Specular Backscatter Detection (CSBD), and Feature Extraction and Classification (FEC) modules as illustrated in Figure 3.

The TDA module in Figure 3 (a) creates MobiDeep DeepFake Detection (MobiDeep-DFD) dataset by collecting and annotating real and fake facial images. The MobiDeep-DFD dataset contains the 4272 annotated corneal-specular reflection segmentation masks for 2136 facial images (two eyes per facial image). 716 real facial images were collected from different datasets, including 565 images from Flickr Faces HQ (FFHQ) dataset [17], 69 images from Celeb-DF dataset, 53 images from FaceForensics++ dataset, and 29 images from DFDC dataset. Similarly, 1420 fake facial images were acquired from various DeepFake detection datasets using various DeepFake generation tools, including 569 face synthesis DeepFake images from StyleGAN2 [15], 431 images from the Celeb-DF dataset, 369 images from the FaceForensics++ datasets, and 51 images from DFDC dataset. As presented in Figure 4, the MobiDeep-DFD dataset contains fake and real facial images in high and low quality with various facial image environmental parameters (FIEP), including illumination conditions, background colors, indoor or outdoor settings, face poses orientations, age, ethnicity, and appearances (e.g., wearing makeup and accessories). As illustrated in Figure 4, the MobiDeep-DFD dataset has two types of annotation for each facial image, including the *Reflection Region Annotation* to define the shapes and locations of *CSB* regions, classifying them into right-reflection and left-reflection classes and the *Image Annotation* to identify the image label (either real or fake), along with *FIEP*, including indoor or outdoor (IO), light or dark (LD), and light strength (LS).

The CSBD module in Figure 3 (b) performs face detection, FIEP feature extraction, and *CSB* localization. CSBD uses a pre-trained MediaPipe Face Detection model to locate a human face in an image and provide its associated position, size, and orientation. CSBD is also responsible for detecting *FIEP*, including indoor or outdoor (IO), light or dark (LD), and light strength (LS). We train a MobileNet-V2 model on the dense indoor and outdoor depth (DIODE) [18] dataset and labeled facial images from the MobiDeep-DFD dataset to classify input images into indoor or outdoor. Our indoor/outdoor dataset includes 20420 images by merging the DIODE and MobiDeep-DFD datasets. To calculate the light strength (LS) of the input facial image, first, we convert the input image color space to LAB format. The *L* channel is independent of color information in the LAB color space and only encodes lightness (intensity). The other two channels *A* and *B* encode color. Next, we extract the *L* channel and normalize it by dividing all pixel values by the maximum pixel value. Finally, it returns the input image's mean of light strength (LS). Analyzing the distribution of the obtained light strength values from our dataset and using the standard deviation, we have a standard way of knowing what image has normal light intensity and which has high light or dark. The input image will be classified as normal light if its mean light strength (LS) is in the range of 0.419 to 0.637, high light if it is more than 0.637, or dark if it is less than 0.419. CSBD also detects right and left *CSB* from the detected face and generates high-quality *CSB* images. We train the CSBD model using the MobileNet-V2 and its modified Single Shot Detector (SSD) version, known as SSDLite, to detect and return the bounding boxes of right and left *CSB* regions and class labels.

Using the right and left *CSB* images extracted from the CSBD module, ***the FEC module in Figure 3 (c)*** performs feature extraction, measures similarity scores, embeds the similarity score with FIEP, and does classification. FEC can extract features from each *CSB* image by using a Siamese Convolutional Neural Network (SCNN) model with various CNN backbones.

1) Feature Extraction: To obtain features from the *CSB* images, the feature extraction model runs a couple of SCNN models in parallel for left and right *CSB*s. The proposed SCNN model consists of two identical CNNs with the same weights to extract deep learning features from the *CSB* inputs. It takes various CNN backbones, including MobileNet-V2, EfficientNet-B0, DenseNet-121, ResNet-152, and VGG-16. We have picked the three most lightweight neural network architectures in both the package size and the number of parameters to cope with the resource constraints (GPU, CPU, memory, and communication) on the mobile devices. Each SCNN module accepts an RGB image of size 224×224 pixels from the CSBD module. Two SCNNs are both used feedforwards to extract features using a global max-pooling layer by removing the fully-connected layer at the top of every network ($include_top$= False). We do not need activation and classes because we only use the backbone models for feature extraction and compare their output at the end by measuring similarity scores.

2) Similarity Measures: As illustrated in Figure 4, *CSB*s are detected in various shapes. According to illumination conditions and background settings, the *CSB*s can be deformed in different colors and blended into the background. For example, in Figure 4, *CSB* shapes of the left and right eyes are different even if the person is looking in the same direction. Also, *CSB*s can be occluded by glasses, eyelids, or eyelashes, and only a tiny portion of reflection can be visible. Hence, the similarity measures of a single factor such as the *CSB* shape or color on both eyes alone cannot be a strong indicator for classifying fake or real images. We measure the similarity scores using the extracted feature vectors, which contain multiple features, including color, edge, and the texture of the *CSB* images. We measure both Euclidean distance scores (EDS) and cosine distance scores (CDS) to statistically compare the similarity between two extracted feature vectors and find the geometric differences between right and left *CSB* images. We applied the ReLU activation function to the EDS and CDS to avoid vanishing gradient problems while training our classifiers. The output [CDS, EDS] generated by SCNN execution represents the semantic similarity between the projected representations of the two input *CSB* images. In addition to the similarity measures, we have designed a feature embedding facility to enhance the feature classification result by applying the environmental factors (FIEP). It is a configurable platform to add multiple embedding functions. For MobiDeep, we have implemented an Environmental Feature Embedding (EFE) function, which takes a few FIEPs, including indoor/outdoor (IO), light/dark (LD), and light strength (LS). As shown in Figure 4, for simplicity, we take boolean values for IO and LD and numerical values from 0 to 1 for LS. These FIEPs can be added and merged according to the requirements. Taking a row of [IO, LD, LS] from the input and annotated FIEP values from the TDA, EFE create adjusted FIEP values such as [IO', LD', LS']. Merging them with the similarity measures [CDS', EDS'] creates a row of 5 col numerical values [CDS', EDS', IO', LD', LS'] as an output. EFE function also takes the right and left CSB features vectors and combines them in one vector for classification.

3) Classification: As illustrated in Figure 3, the classification module finally classifies images to either real or fake by taking a row of 5 column values [CDS', EDS', IO', LD', LS'] created from the EFE function. We defined the classification network with a sequence of five blocks. The first block consists of a single BatchNormalization layer that normalizes its inputs by applying a transformation that maintains the mean output close to 0 and the output standard deviation close to 1. The following three blocks are similar. Every block consists of a sequence of a fully connected (*fc*) layer with 128 nodes, a single BatchNormalization layer followed by a ReLU activation function. The BatchNormalization layer centers the learned features from the fully connected layer on 0, while the ReLU activation uses 0 as a pivot to keep or drop the activated channels [19]. The fifth block consists of two layers, a concatenate layer to merge the fourth block's output tensor with the right and left CSB features tensor, and a fully connected layer (predication layer) with two nodes and a softmax activation function to return a probability distribution for binary classification. A binary cross-entropy probabilistic loss function is used to compute the cross-entropy loss between actual labels and predicted labels and measure how accurate the model is during training and testing. Eventually, it creates a binary classification result (either real or fake) as an output.

IV. EVALUATIONS

We conducted extensive experiments on the MobiDeep implementation in Android, iOS, and web applications to evaluate the performance under real-world scenarios and compare the accuracy and speed with current state-of-the-art (SOTA) DeepFake detection methods.

A. Evaluations of Execution Speed

The primary goal of the experiments is to assess the feasibility of MobiDeep usage on mobile devices by checking the performance of MobiDeep classification speed on both GPU and CPU environments and find suitable feature extractor models among MobileNet-V2, EfficientNet-B0, and DenseNet-121 for the mobile application.

Figure 5 shows the testing speed on the GPU environment using the Google Colab Compute Engine (GCE) VM backend (with NVIDIA Tesla-P100-PCIE-16GB) and 8-cores CPU. Testing batch sizes (i.e., images per step) increases, and the delay for all models on both GPU and CPU increases. MobileNet-V2 is the fastest, and DenseNet-121 is the slowest. As presented in the left-hand panel of Figure 5 for the typical batch size of 128 images with GPU, all models can evaluate within 250 ms. In contrast, the right-hand panel of Figure 5 shows that with 8-core CPU, MobileNet-V2 and EfficientNet-B0 both models can classify a batch size of 128 images in 3 seconds and 6 seconds, respectively. The DenseNet-121 delay grows faster than other models on an 8-core CPU. MobileNet-V2 offers the fastest evaluation speed.

TABLE I
DETECTION SPEED ON ANDROID AND IOS.

Backbones	**Type**	**FEC Delay (ms)**	**Total Delay (ms)**
MobiDeep (DenseNet-121)	Galaxy S9	69.68	191.31
	iPhone 11	72.83	197.36
MobiDeep (MobileNet-V2)	Galaxy S9	45.82	167.45
	iPhone 11	47.59	172.12

As shown in Table I, we assessed the feasibility of MobiDeep on mobile devices, including Android and iOS. We have built a real-time, cloudless, lightweight cross-platform mobile application using React Native user interface software framework and TensorFlow.js hardware-accelerated JavaScript library for deploying our ML models on mobile devices. Samsung Galaxy S9 (SM-G960F) comes with a 2.7 GHz Octa-Core processor, 64 GB memory, 4 GB RAM, and a 3000 mAh battery. iPhone 11 has an A13 Bionic chip (with 6-core CPU, 4-core GPU, and 8-core Neural Engine), 128 GB memory, 4 GB RAM, and a built-in rechargeable lithium-ion battery. We collected the average execution speed of CSBD and FEC with two deep neural network feature extraction architectures (MobileNet-V2 and DenseNet-121). MobiDeep (MobileNet-V2) has a low average execution delay on both Samsung Galaxy S9 (167.45 ms) and iPhone 11 (172.12 ms) compared to MobiDeep (DenseNet-121)'s average execution speed was (191.31 ms) on Samsung Galaxy S9 and (197.36 ms) on iPhone 11. MobiDeep operates efficiently within 200 ms on Android and iOS mobile devices with an easy-to-use, stand-alone, and lightweight mobile application.

B. Classification Using Different Backbone Models for Feature Extraction

The primary goal of the experiments is to assess the feasibility of MobiDeep usage on mobile devices by checking the performance of MobiDeep classification accuracy with various feature extractor models. As shown in Table II, we used the three most lightweight CNN backbones (MobileNet-V2, EfficientNet-B0, and DenseNet-121) for feature extraction. Three classifiers were trained on the MobiDeep-DFD training dataset and tested on the MobiDeep-DFD testing dataset. Table II shows that classifier accuracy with different feature extractors. MobiDeep is highly effective (over 90%) in

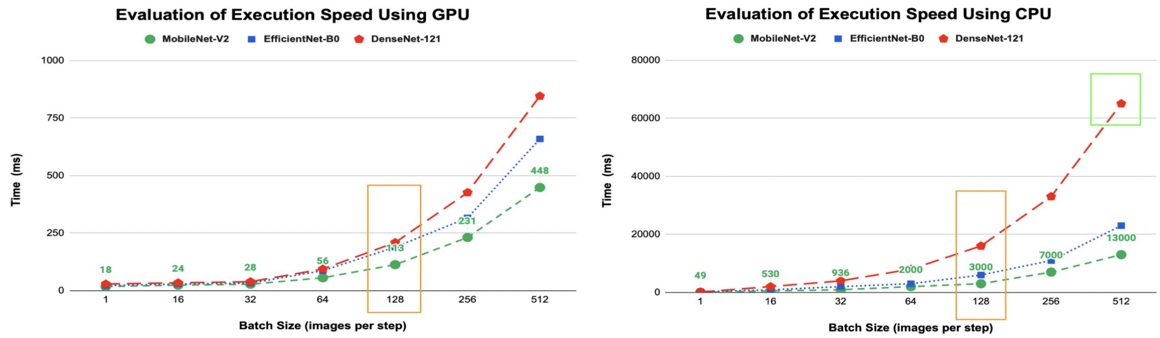

Fig. 5. Evaluation of Testing Speed with GPU and CPU using Different Backbone Models for Feature Extraction.

TABLE II
CLASSIFICATION ACCURACY.

Backbones	Accuracy	Loss
MobiDeep (EfficientNet-B0)	91.27	0.185
MobiDeep (DenseNet-121)	97.35	0.053
MobiDeep (MobileNet-V2)	**98.70**	**0.029**

detecting DeepFake images. MobiDeep (MobileNet-V2) is the best in both accuracy (98.70%) and loss (0.029). MobiDeep (DenseNet-121) is the second-best in both accuracy (97.35%) and loss (0.053). Hence, MobileNet-V2 and DenseNet-121 can be used for feature extraction without any significant difference. However, MobiDeep (EfficientNet-B0)'s accuracy is the least (91.27%), and MobiDeep (EfficientNet-B0)'s loss is the highest (0.185). Hence, EfficientNet-B0 may not be recommended for MobiDeep's feature extraction.

V. CONCLUSIONS

This paper presented the design and development of a real-time, cloudless, lightweight mobile DeepFake detection technology named **MobiDeep** (**Mobi**le **Deep**Fake Detection through Machine Learning-based Corneal-Specular Backscattering). Focusing on the hypothesis that the existing DeepFake methods, including replacement, editing, and synthesis, are hard to coordinate their counterfeits with the reflective elements, MobiDeep took a novel approach using the corneal-specular backscatter images of human eyes. It evaluates the similarity and consistency with multiple surrounding environment features, Facial Image Environmental Parameters (FIEP), including color components, shapes, and textures, instead of merely checking the similarity between eye reflection shapes. We have implemented a cross-platform mobile application to evaluate the performance using various input parameters and lightweight Deep Neural Network (DNN) architectures. The experimental results show that MobiDeep achieved the high accuracy (98.70%) and rapid detection speed (less than 200 ms) in detecting sophisticated DeepFake images using MobileNet-V2 with the MobiDeep-DFD dataset.

REFERENCES

[1] Reface, "Reface," 2021. [Online]. Available: https://reface.app
[2] Avatarify, "Avatarify: AI Face Animator App," 2021. [Online]. Available: https://avatarify.ai
[3] Wombo, "Wombo," 2021. [Online]. Available: https://www.wombo.ai
[4] H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu, "Multi-attentional deepfake detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2185–2194.
[5] H. Guo, S. Hu, X. Wang, M.-C. Chang, and S. Lyu, "Eyes tell all: Irregular pupil shapes reveal gan-generated faces," 2021.
[6] Z. Sun, Y. Han, Z. Hua, N. Ruan, and W. Jia, "Improving the efficiency and robustness of deepfakes detection through precise geometric features," 2021.
[7] H. Liu, X. Li, W. Zhou, Y. Chen, Y. He, H. Xue, W. Zhang, and N. Yu, "Spatial-phase shallow learning: Rethinking face forgery detection in frequency domain," 2021.
[8] S. Hu, Y. Li, and S. Lyu, "Exposing gan-generated faces using inconsistent corneal specular highlights," *arXiv preprint arXiv:2009.11924*, 2020.
[9] Y. Li, M.-C. Chang, and S. Lyu, "In ictu oculi: Exposing ai generated fake face videos by detecting eye blinking," *arXiv preprint arXiv:1806.02877*, 2018.
[10] F. Matern, C. Riess, and M. Stamminger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*. IEEE, 2019, pp. 83–92.
[11] Yonhap News Agency, "KAIST unveils deepfake-detecting mobile app," Mar. 2021. [Online]. Available: https://en.yna.co.kr/view/AEN20210330004200320
[12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.
[13] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.
[14] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.
[15] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," 2020.
[16] Icons8, "Face Swapper," 2022. [Online]. Available: https://icons8.com/swapper
[17] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4401–4410.
[18] I. Vasiljevic, N. I. Kolkin, S. Zhang, R. Luo, H. Wang, F. Z. Dai, A. F. Daniele, M. Mostajabi, S. Basart, M. R. Walter, and G. Shakhnarovich, "DIODE: A dense indoor and outdoor depth dataset," *CoRR*, vol. abs/1908.00463, 2019. [Online]. Available: http://arxiv.org/abs/1908.00463
[19] F. Chollet, *Deep learning with Python*. Simon and Schuster, 2021.