ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

PRESENTATION ATTACK DETECTION WITH SHUFFLED PATCH-WISE BINARY SUPERVISION

M.Sc. THESIS

Alperen KANTARCI

Department of Computer Engineering

Computer Engineering Programme

JUNE 2022

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

PRESENTATION ATTACK DETECTION WITH SHUFFLED PATCH-WISE BINARY SUPERVISION

M.Sc. THESIS

Alperen KANTARCI
(504191504)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Prof. Dr. Hazım Kemal EKENEL

JUNE 2022

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

KARIŞTIRILMIŞ YAMA TABANLI İKİLİ GÖZETİM İLE SUNUM SALDIRI TESPİTİ

YÜKSEK LİSANS TEZİ

Alperen KANTARCI
(504191504)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Hazım Kemal EKENEL

HAZİRAN 2022

Alperen KANTARCI, an M.Sc. student of ITU Graduate School student ID 504191504 successfully defended the thesis entitled "PRESENTATION ATTACK DETECTION WITH SHUFFLED PATCH-WISE BINARY SUPERVISION", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Hazım Kemal EKENEL** Istanbul Technical University	.....
Jury Members :	**Prof. Dr. Gözde ÜNAL** Istanbul Technical University	.....
	Prof. Dr. Lale AKARUN Boğaziçi University	.....
		.....

Date of Submission : **03 June 2022**
Date of Defense : **22 June 2022**

To my family,

FOREWORD

My journey with academia started in 2018 when I first met with the SiMiT Lab. and Prof. Hazım Kemal Ekenel. That first meeting changed my life entirely, and it led me to pursue a research career in academia, therefore led me to write this dissertation. I want to thank Prof. Hazım Kemal EKENEL for his unconditional support, guidance, patience, and inspiration for my academic research during my Bachelor's and Master's study. He has always inspired me as a researcher, professor, and person.

As much as we get guidance from our advisors, the people with whom we work together are the ones that teach us the most. Therefore I would like to thank to Deniz Engin, Doğucan Yaman, İrem Eyiokur Yaman, Azmican Özgen, and Omid Abdollahi Aghdam. I learned a lot from them. I want to express my special thanks to Şeymanur Aktı for her technical and moral support during my Master's journey. She and I shared a lot while working, and she helped me a lot to carry my burden of work. I feel grateful to work with her as a colleague. Apart from all, I would also like to thank Ali Eren Çelik, Safa Keskin, Gamze Akyol, Burak Bekci, Ahmet Faruk Bayrak, Ali Azmoudeh, and all of my dearest friends for their technical and moral support. I would not be able to work in these stressful times without their friendship.

This thesis is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) under the 2210-A National MSc Scholarship Program. I would like to thank both TUBITAK and Sodec Technologies for their support of this thesis.

Last but not least, none of my work would have been possible without the support of my family. In every stage of my education, my family always supported me much more than any family would. I always felt their presence on my back, no matter what I do. Thanks to their support, I have never worried about anything except my education. I want to express my utmost gratitude and appreciation to my father, mom, and beloved sister. They are the hidden authors of this thesis, and they deserve all the compliments.

June 2022

Alperen KANTARCI
Computer Engineer

TABLE OF CONTENTS

ABBREVIATIONS

CNN : Convolutional Neural Network
ANN : Artificial Neural Network
App : Appendix
BP : Backpropagation
PA : Presentation Attack
PAD : Presentation Attack Detection
PAI : Presentation Attack Instrument
RNN : Recurrent Neural Network
APCER : Attack Presentation Classification Error Rate
BPCER : Bona Fide Classification Error Rate
ACER : Average Classification Error Rate

SYMBOLS

y	**:** Image Label
log	**:** Logarithm
X	**:** Input Image
s	**:** Patch Size
$\mathbf{p_y}$	**:** Patch Label

LIST OF TABLES

Page

LIST OF FIGURES

Page

PRESENTATION ATTACK DETECTION WITH SHUFFLED PATCH-WISE BINARY SUPERVISION

SUMMARY

Face recognition systems have been one of the most commonly used biometrics in various different applications, such as mobile payments, smart phone security, and accessing to high-security areas. However, face presentation attacks created by people who obtain biometric data covertly from a person or through hacked systems are among the major threats to face recognition systems. Presentation attacks are easy to make, especially for face presentation attacks, as malicious individuals only need a high-quality face image of any enrolled user to bypass the biometric system.

In order to detect these attacks, Convolutional Neural Network (CNN) based systems have gained significant popularity recently. Convolutional Neural Networks provide end-to-end systems for presentation attacks. They also offer fast inference, which is helpful for the biometric systems. However, CNN-based systems need a substantial amount of data to train. It is hard to acquire presentation attacks as for each attack, a human should physically attack to the sensor. Unlike face recognition datasets that utilize millions of face images crawled from the internet, presentation attacks have to be captured explicitly for the dataset. Therefore, publicly available datasets are significantly smaller. As neural networks require massive amount of data to generalize, CNN-based presentation attack detection systems perform very well on intra-dataset experiments. Yet, they fail to generalize to the datasets that they have not been trained on. This indicates that they tend to memorize dataset-specific spoof traces. To mitigate this crucial problem, we propose a new presentation attack detection training approach that combines pixel-wise binary supervision with patch-based Convolutional Neural Networks. We call our method as Deep Patch-wise Supervision Presentation Attack Detection (DPS-PAD). The proposed method combines different patches of different attacks and bona fide images of the dataset and creates a new training data this way.

Our experiments show that the proposed patch-based method forces the model not to memorize the background information or dataset-specific traces. We extensively tested the proposed method on widely used presentation attack detection datasets —Replay-Mobile, OULU-NPU— and on a real-world dataset that has been collected for real-world presentation attack detection use cases. The proposed approach is found to be superior in challenging experimental setups. Namely, it achieves higher performance on OULU-NPU Protocol 3, 4 and on inter-dataset real-world experiments.

KARIŞTIRILMIŞ YAMA TABANLI İKİLİ GÖZETİM İLE SUNUM SALDIRI TESPİTİ

ÖZET

Günümüzde birçok sistem kullanıcılarının hesaplarını ve yetkilerini güvende tutabilmek için biyometrik verilerin doğrulanmasını istemektedir. Pek çok yüksek güvenlikli bina, bankacılık sistemleri, sağlık uygulamaları ve hatta kişisel akıllı telefonlar kullanıcı girişi için ekstra olarak biyometrik verileri kullanmaktadır. Geleneksel olarak kullanılan PIN kodları veya şifreler gibi çalınması kolay olmayan ve her kişinin kendine özel ve biricik olan biyometrik veriler ile yapılan doğrulamalar hem daha güvenli hem de kullanıcılar için daha kolay bir erişim yöntemidir. Biyometrik verilere parmak izi, avuç içi izi, kulak, ses, yüz, iris, retina, imza, vücut kokusu, el damarı veya DNA bilgisi örnek olarak verilebilir. Fakat çok sayıdaki bu verilerin alınması için farklı donanımlar gerekmektedir. Örnek olarak, parmak izi almak için gereken sensörler, ışıklandırma ve parmağın sensöre nasıl yerleştirildiği gibi teknik zorluklar her biyometrik verinin günlük yaşamdaki doğrulamalar için kullanılamamasına sebep olmaktadır. Bu biyometrik verilerden en kolay elde edilebilen ve kullanılabilenlerin başında yüz gelmektedir.

Yüz biyometrisini elde etmek için gerekli olan tek sensör kameradır. Günümüzde maliyetleri oldukça düşen kameralar, zaten birçok yerde kullanıldığı için biyometrik doğrulama için de sıklıkla kullanılmaktadır. Biyometrik doğrulamanın dışında yüz biyometrisi askeriye, güvenlik güçleri, sağlık uygulamaları, kayıp insanların tespiti gibi birçok alanda da karşımıza çıkmaktadır. Bu alanlarda kişi tanılama, kişi doğrulama ve kişi onaylama gibi farklı amaçlarla kullanılan yüz biyometrisi hem sağlayabileceği başarı oranı hem de sağladığı güvenlik seviyesi nedeniyle hem akademide hem de endüstride üstüne oldukça çalışılmaktadır. Fakat yüz biyometrisi kullanan bu sistemlerin en büyük zayıflığı verilen biyometrik verinin gerçekten kişinin kendisinden gelip gelmediğinin tespit edilmesinin zor olmasıdır.

Bir biyometrik sistemin sensörüne, bir kullanıcının çalınmış verilerinin sunulmaya çalışıldığı saldırılara sunum saldırıları adı verilmektedir. Sunum saldırıları her türlü biyometrik sistemde olabilse de bu tez kapsamında yüz sunum saldırılarının tespiti konusu incelenmektedir. Yüz sunum saldırıları günümüzde daha çok sisteme kayıtlı kullanıcıların yüz imgelerinin sosyal medya platformlarındaki profil imgelerinden kopyalanması veya kişinin yüz imgelerinin farklı yollardan ele geçirilmesi ile yapılmaktadır. Ele geçirilen yüz imgesi bir kağıda basılabildiği gibi bu imge üzerinden silikon veya reçine maskeler oluşturularak biyometrik sistem kandırılmaya çalışılabilmektedir. Buna ek olarak dijital ortamlardan elde edilen imge ve videolar telefon, tablet, televizyon, monitör gibi görüntüleme cihazlarına aktarılıp sunulan biyometrik verinin gerçekçiliği artırılabilmektedir. Günümüzde birçok biyometrik sistem sadece tek bir imge değil, video kaydı ile sunulan biyometrik verinin

gerçekçiliğini daha kolay bir şekilde tespit etmeye çalışmaktadır. Fakat tek bir imgeden sunum saldırı tespiti yapmanın oldukça başarılı olduğu gösterilmiştir.

Artan veri miktarı, ucuzlayan donanım maliyetleri ve sinir ağları mimarilerinde bulunan yeni yöntemler sayesinde derin öğrenme modelleri geleneksel yöntemlerden daha başarılı şekilde bilgisayarlı görü, ses tanıma, doğal dil işleme, medikal görünteleme ve robotik gibi birçok alana adapte edilmiştir. Yüz sunum saldırı tespitleri için geleneksel öznitelikleri kullanan sistemler sunulmuş olsa da son yıllarda hızlı bir şekilde kabiliyetlerini kanıtlayan yapay sinir ağları sunum saldırı tespiti için de kullanılmaya başlamıştır. Yüz sunum saldırıları özünde ikili sınıflandırma olarak formülize edilmektedir. Pozitif sınıf sensörün karşısında gerçek ve canlı bir insan olduğunu ve verilen yüzün herhangi bir cansız materyalden gelmediğini belirtirken, negatif sınıf ise verilen imge veya videodaki yüzün canlı bir insana ait olmadığını belirtir. Sunum saldırı tespiti problemi ayrıca yüz canlılığı tespiti olarak da adlandırılmaktadır. Fakat buradaki canlılık kavramı kafa karıştırıcı olabilmektedir. Canlılık kavramı ile kastedilen sensörün karşısındaki kişinin canlı olması ve yüzünün herhangi bir maske, kağıt, elektronik cihaz veya makyaj gibi yüz biyometrisini değiştirecek objelerle kapatılmamış olmasıdır.

Yapay sinir ağlarının yüksek başarım göstermesi ve genelleştirilebilir sonuçlar verebilmesi için çok fazla sayıda veriye ihtiyacı bulunmaktadır. Sunum saldırı tespitine en yakın alanlardan biri olan yüz tanıma alanındaki yapay sinir ağları için milyonlarca imge kullanılmaktadır. Fakat sunum saldırı veri kümelerini oluşturmak diğer bilgisayarlı görü alanlarındaki kadar kolay olmamaktadır. Her bir saldırı için bir kişinin yüz imgesini veya videoyu içeren kağıt, maske, görüntüleme cihazı gibi saldırı enstrümanını oluşturması ve bu enstrümanla birlikte sistemi olabildikçe kandırabilecek bir şekilde sensöre karşı tutması gerekmektedir. Video tabanlı saldırılar için ise belli bir süre sensör karşısında hareket etmek gerekmektedir. Bu zaman alan ve oldukça maliyetli olan saldırı görüntülerinin toplanması kolay olmadığı için herkese açık olarak sunulan sunum saldırı tespiti veri kümeleri hem veri sayısı hem de kişi sayısı bakımından oldukça sınırlı kalmaktadır. Bu sınırlı veri kümeleri ile eğitilen yapay sinir ağı tabanlı sunum saldırı tespiti sistemleri ise az sayıda verilen veriyi çok iyi öğrenirken, veri kümesi dışından gelen saldırılara başarılı karşılık verememektedir. Özellikle sunum saldırı tespiti alanında bu aşırı öğrenme problemi, çözülmeye çalışılan en büyük sorunlardan biri olarak karşımıza çıkmaktadır. Kısıtlı veri ile eğitilen sistemlerin bir diğer sorunu ise daha önce görülmemiş bir saldırı çeşidine karşı cevap verememektir. Örneğin sadece belli bir silikon maske tipiyle oluşturulmuş 3 boyutlu maske saldırılarını içeren bir veri kümesinde eğitilen bir sistem, reçineden veya 3 boyutlu yazıcıdan çıkartılmış bir maske saldırısına iyi cevap verememektedir.

Sunum saldırı tespitinde var olan bu aşırı öğrenme ve genelleştirilememe problemini çözmek için sayısız çalışma yapılmıştır. Veri kümelerinin sayılarını artırmak zor ve maliyetli olduğundan dolayı daha çok yapay sinir ağlarının geliştirilmesine odaklanılmıştır. Bu tezde sunulan yöntem de aynı soruna farklı bir açıdan bakarak, elde bulunan verilerin en iyi şekilde sentezlenerek eğitilen modellerin daha genelleştirilebilir olmasını sağlamaktadır. Alandaki birçok çalışma modellerini eğitirken bir yüz imgesi ve bu imgenin ait olduğu sınıf etiketini kullanarak ikili sınıflandırma yapmaktadır. Bu çalışmaların aksine ikili sınıflandırmayı piksel

seviyesinde yapan DeepPixBis adlı yapay sinir ağı ise verilen imgenin sınıfını piksel seviyesinde göstererek, sunum saldırı tespiti başarımlarını oldukça yükseltmiştir. Bu tezde de DeepPixBis modeline verilen girdi imgeleri ve etiketler değiştirilerek eğitim yapılacak imge sayısı artırılmaya çalışılmış ve önerilen yöntemin genelleştirmeyi ciddi bir düzeyde artırdığı gösterilmiştir.

Bu tezde sunduğumuz yöntem verilen sunum saldırı veri kümesindeki bulunan tüm yüzleri ufak yamalara ayırarak yama tabanlı bir tespit yapmaya çalışmaktadır. Her bir yüz imgesi 49 tane küçük yamaya bölündükten sonra her bir yama için imgenin etiketi, yamanın etiketi olarak atanmaktadır. Daha sonra alandaki diğer çalışmalardan farklı bir şekilde hem saldırı hem de gerçek yüz yamaları tek bir imge haline getirilmek üzere karıştırılmıştır. Bu işleme yama dikimi adı verilmekte ve yama dikimi iki farklı şekilde gerçekleştirilebilmektedir. Birinci yöntem yüz şeklini korurken, ikinci yöntemde yüz şeklini dikkate almadan dikim gerçekleştirilir. Yüz şeklini koruyarak yapılan dikim işleminde farklı kişilerin hem saldırı hem de gerçek yamalarının alındığı yüz parçası dikkate alınarak birleştirildiği için, dikim işlemi sonunda ortaya yüzü andıran bir imge çıkmaktadır. Lakin bu imge 49 farklı yüz yamasından oluşmaktadır, bu yüzden de içinde hem gerçek hem saldırı imgelerinin parçalarını içeren bir karışımdır. İkinci yöntem olan rastgele dikimde ise herhangi bir yüz şekli dikkate alınmadan rastgele olarak dikilmektedir.

Dikim işleminin modele sağladığı katkı ise, yapay sinir ağının spesifik sunum saldırı izlerine odaklanmayıp verilen küçük yüz yamalarından bile sunum saldırısı olup olmadığını öğrenebilmesidir. Bunun yanında her yüzden 49 yama alındığı ve bu yamalar tamamen rastgele olarak birleştirildiği için çok yüksek sayıda eğitim verisi oluşturulabilmektedir. Böylece kısıtlı imge sayısı bulunan veri kümelerinin üzerinde bile eğitim sağlamak mümkün olabilmektedir. Ayrıca yüz yapısını bozarak yapılan eğitimlerin farklı veri kümelerinde yapılan deneylerde performansa katkı sağladığı gösterilmiştir.

Alanda kabul görmüş test veri kümeleri üzerinde yapılan deneyler sonucunda önerdiğimiz DPS-PAD (Derin Yama Gözetimli Sunum Saldırı Tespiti) yöntemi sadece girdi imgelerinin farklı yamalardan birleştirilmesi sayesinde verilen herhangi bir sunum tespit sisteminin başarısını dikkate değer şekilde artırmıştır. Ayrıca önerilen yöntem herhangi bir yapay sinir ağı tabanlı modele uygulanabildiği için de oldukça kullanışlıdır. Alandaki kabul görmüş test veri kümeleri daha çok laboratuvar ortamında kontrollü olarak oluşturulduğu için, günlük hayattaki şartları taklit etmeye çalışan ve kontrolsüz ortamdan toplanan verilerle Sodec Real-World veri kümesi oluşturulmuştur. Yapılan deneylerle beraber sunduğumuz yöntemin daha çok genelleştirmeye ihtiyaç duyan kontrolsüz ortamdaki verilerle test edildiği başarılı olduğu gösterilmiştir. Bu testlerde alandaki en iyi performanslı modellerden daha başarılı olduğu gösterilmiştir.

1. INTRODUCTION

Face recognition technology has been a popular research field in both industry and academia due to its ease of use and availability of cheap camera sensors. The technology has been deployed for many end devices, high-security areas, mobile payment systems, and border control [1,2]. Therefore, it has been a major target of people with malicious intentions to use the face of an enrolled user, which can be accessed easily, to gain authorization to systems protected by biometric access. Face images of legitimate users can be downloaded easily due to the widespread usage of social networks and search engines. Typing a person's name and downloading their profile pictures are enough to steal a person's facial biometric data. Downloaded face images can be printed on paper, silicon mask, or resin mask or can be shown on high-quality display devices, such as laptop screens and televisions. These kinds of attacks are called presentation attacks (PA). Figure 1.1 shows an example presentation attack scenario.

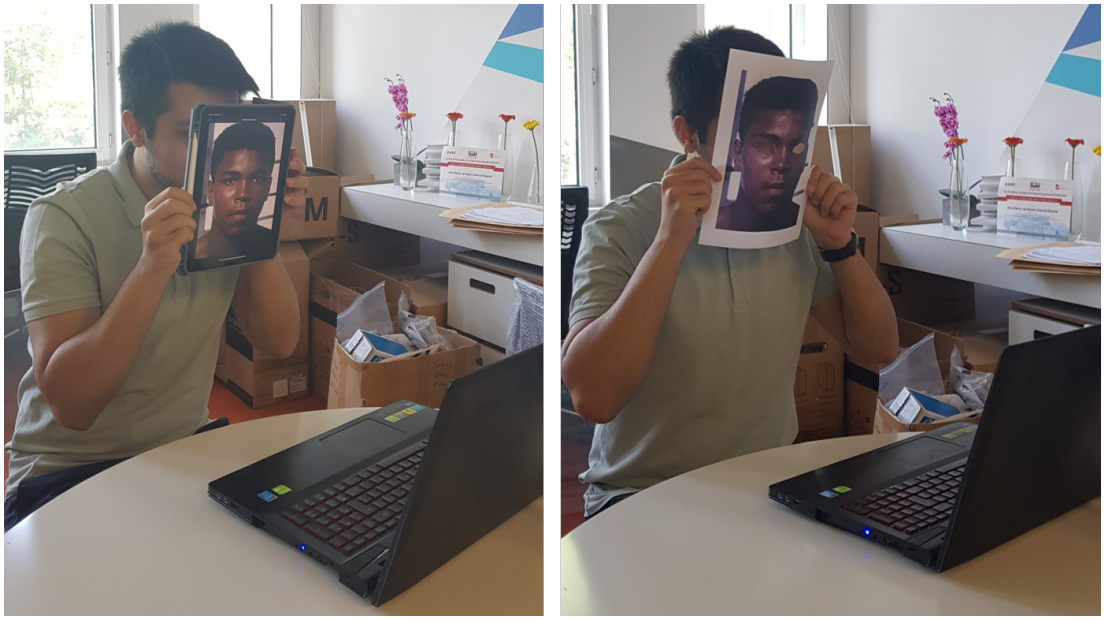


Figure 1.1 : An example presentation attack scenario where impostor uses tablet and paper as presentation attack instrument.

In general, ISO/IEC IS 30107 [3] defines the presentation attack as "a presentation to the biometric capture subsystem with the goal of interfering with the operation of the biometric system". Systems that aim to detect presentation attacks are called presentation attack detection (PAD) systems. Presentation attacks are not limited to face biometrics but in this thesis, we are focusing on face presentation attacks. Other biometric data types for presentation attacks are iris [4], fingerprint [5], and signature [6].

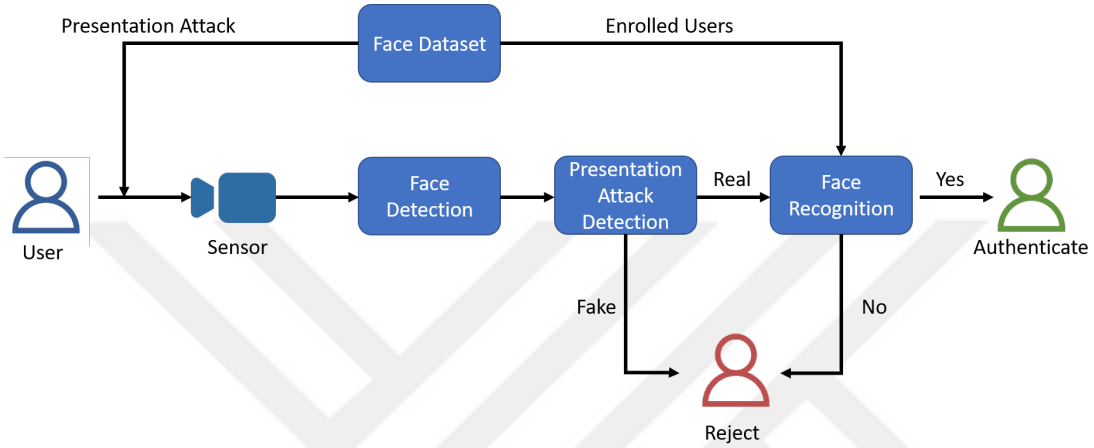


Figure 1.2 : An overview of face recognition pipeline for a biometric authentication system. Presentation attacks targets biometric capture device, where given biometric data is interfered by the impostor or identity concealer.

Figure 1.2 shows general usage of face recognition systems. Presentation attack detection takes place before the face recognition to determine that given biometric data was captured from a real person. Therefore, presentation attacks are direct attacks on the capture devices, therefore named sensor level attacks. PAD is also called liveness detection, however liveness keyword might be confusing for the attacks that contain transparent or half masks. Therefore, the "liveness" is measured by assessing whether any presentation attack instruments are present in the image. These instruments can be papers, tablets, mobile phones, laptop screens, televisions, transparent masks, silicone masks, resin masks, or even makeups. Figure 1.3 shows some examples of these attacks. These instruments are grouped into paper, replay, and mask attacks. In paper and replay attacks, the presented face images are not live as they are either on a paper sheet or displayed on a display monitor. In mask attacks, the impostor or identity concealer wears a mask that covers some parts of a face. Therefore, the person is "live"

but their faces are covered with different kind of materials such as resin or silicone. If the PAD fails to detect the attack, face recognition systems are vulnerable as the given data belongs to an enrolled user. It is a challenging problem even for humans, as presentation attack instruments (PAI) can be very high quality such as 8K televisions or silicone masks. Frame-based PAD is even a more challenging problem as no temporal information can be used to detect attack traces and abnormalities.

In the earliest works on PAD, researchers created hand-crafted features to use the color and texture of the given face images to detect liveness. PAD systems gained a significant performance increase after using deep learning methods, instead of hand-crafted features. Even after these improvements, PAD methods are not able to generalize well to more realistic scenarios. Deep learning methods need millions of data in order to train models that can detect presentation attacks from live recordings. However, collecting PAs is highly time-consuming and costly as high-resolution displays or expensive face masks are used as PAIs. Many academic research laboratories have collected large and representative datasets [7]–[12] to train deep learning models and to create a benchmark to test PAD system performances fairly. As many of these datasets are collected under controlled environments to properly test the developed systems, they might not reflect real-world presentation attacks. Rapid performance increases have been achieved in PAD task when researchers found different cues to decide the liveness of a face [13]–[15]. These different cues are generally effective to prevent specific attack types, but cannot perform well on all kinds of attacks like mask, replay, and print attacks. Deep neural networks have been trained with the data to learn these different cues to create PAD systems that work very well on intra-dataset benchmarks. Yet, the real challenge in PAD still remains as achieving high performance on inter-dataset benchmarks, which shows the real performance of the PAD systems in real-world like scenario. Most of the systems that use CNNs overfit the data easily by memorizing reflection and illumination effects. Our aim in this paper is using patch-wise supervision to reduce this overfit behaviour and increase the generalization capability of PAD systems.

In this thesis, we present a novel training procedure, namely Deep Patch-Wise Supervision for Presentation Attack Detection (DPS-PAD), for face PAD systems.

We also present comprehensive analysis of our novel method by investigating the effects of different parts of the procedure. Our proposed strategy leverages small face patches, therefore suitable for using less number of images. As PAD datasets do not contain large amount of images to train a well generalized deep learning models, our model is preferable. We show that our training method, which combines different face patches, utilizes the pixel-wise binary loss better. We also show that our proposed method improves model performances in real-world experiments. To assess the real-world performance, we use a dataset that we collected specifically for this purpose. The uncontrolled nature of the dataset imitates real-world data with the unrestricted presentation of attack instruments and input sensors.

The remainder of the thesis is organized as follows: In Chapter 2, we provide a brief overview of the related work. In Chapter 3, we explain the proposed method and give detailed information about the training parameters. In Chapter 4, we present both publicly available benchmark datasets and the dataset that we have collected to imitate the real-world conditions. We use these datasets in our experiments and the ablation study of the proposed method. This section also provides experiment results. In Chapter 5, we present more insight into the method and discuss its limitations. Finally, Chapter 6 concludes the thesis and provides future directions for our research.

Figure 1.3 : Examples of different presentation attack instruments. Top row shows cut paper, eye mask, and laptop screen. Bottom row shows tablet, display monitor, and silicone mask.

2. LITERATURE REVIEW

In this chapter, we present the related literature review for presentation attack detection.

PAD methods can be grouped into two main approaches; video-based and frame-based. Frame-based methods use a single image to detect presentation attacks and leverage subtle cues from the given face image. On the other hand, video-based methods [16]–[18] additionally employ temporal consistency and temporal cues. These cues can be summarized as liveness, texture, and 3D geometry [19]. Among these cue-based methods, liveness cues are not suitable for frame-based PAD methods. Therefore, texture and 3D geometry cue-based methods [20]–[23] are more popular for detecting liveness from a frame. Depth or pseudo-depth signals are the most popular signals for 3D geometry cue-based methods to distinguish presentation attacks from real attempts. Even though devices with depth sensors, time-of-flight cameras, Lidar sensors, etc., are getting popular in daily usage, most mobile phones or video cameras do not have depth sensors. Therefore, methods mostly rely on pseudo-depth maps which are not real data and may not reflect real-world data distribution very well. For that reason, most of these methods might get good results on specific datasets but fail to generalize.

As one of the initial PAD works with deep neural networks, [24] proposes to use a face alignment network on face images and use those images to train an AlexNet [25] model. The trained model is utilized as a facial feature extractor network. Extracted features then classified as attack or bona fide by using a support vector machine classifier.

In order to improve PAD performances, researchers utilize secondary supervision, along with the binary classification objective, for training their models with a limited amount of data. [26] provides an exceptional overview on deep learning-based PAD methods. Among these methods, [27] proposes an effective model for frame-level PAD. They add additional supervision, which they call pixel-wise binary supervision,

to simplify the necessity of complex depth maps and temporal information. Their model creates a 14×14 score map which helps to perform pixel-wise binary supervision. On top of this supervision, their model is guided with binary cross-entropy. We build our model on top of [27] by using their pixel-wise binary supervision and model architecture. Instead of using binary cross-entropy, we propose to use only pixel-wise binary supervision. Moreover, we train our models with shuffled face images that are created by multiple patches of different face parts of different subjects.

There has been more follow-up work on pixel-wise supervision due to its outstanding performance on benchmarks. [28] provides a good overview of pixel-wise supervision, along with depth maps, 3D point clouds, and analysis of different loss functions for PAD. They also propose a pyramid supervision, inspired by the [29] to supervise the network using both local spatial relations and high-level semantics. For the pyramid supervision, they both consider depth map pixel-wise supervision and binary pixel-wise supervision. They show that using pyramid supervision with pixel-wise supervision enhances the model's interpretability and performance on various benchmarks. [30] proposes to use pixel-wise supervision with the meta-teacher model to supervise presentation attack detectors. They show that meta-teacher model produces more informative supervision than using a fully supervised training or teacher-student models. Their findings indicate that the proposed model can provide sufficient supervision to train networks without depending on highly time-consuming pixel-wise labels. Another work that uses meta-learning is [31], which formulates the PAD problem as zero- and few-shot learning tasks. They train a meta-learner that focuses on detecting unseen presentation attack types by using live and spoof videos and a few examples of new attacks. They show that their method can outperform the others on existing zero-shot PAD benchmarks.

[15] proposes a two-stream CNN that uses patches and depth maps. They claim that patch-based CNN learns to discriminate artefact patches independent of the spatial face areas whereas depth-based CNN allows the model to learn how a face-like holistic depth map should look like. Our work, which combines pixel-wise binary supervision with patch-based CNN, is inspired by [15,27]. As [15] showed, we believe that

training a CNN with face patches allows the model to distinguish artefacts without learning background or dataset-specific traces. Therefore, it prevents the model to overfit. Effectiveness of a similar patch shuffling method is presented in [32]. In [32] it is shown that shuffling pixels within a patch can increase the generalization of the models. Our approach tries to mimic this behavior from a higher level of view. Currently, one of the best performing frame-level PAD methods [33,34] leverages a novel convolution operation. Authors propose Central Difference Convolution (CDC) for detecting presentation attack artefact traces. CDC specifically focuses on these artefact traces. The model is trained with pseudo-depth maps which require additional pseudo-depth map creation steps for ground truth. They also use computationally expensive Neural Architecture Search (NAS) to find a better and more efficient model which they call CDCN++. They report the lowest error rates on the OULU-NPU [7] dataset. Another work that uses CDC is [35], which claims that aggregating central difference cues make the CDC redundant in the training phase. Therefore, in [35] they propose using two Cross Central Difference Convolutions (C-CDC) which substitutes center and neighbors. Their new convolution method provides lower computation costs. Moreover, they also propose a patch exchange augmentation method to increase the number of images and diversity in the PAD datasets. Similar to our proposed patch stitching method, they propose using different real and spoof face patches by exchanging patches between images. Unlike our binary pixel-wise supervision, they use depth maps, therefore exchanged patches also change the depth map labels. Facial structure during the patch exchange is preserved similar to our controlled-stitching method.

Recently, there have been a couple of works that propose to use local patches of the face with deep learning architectures. Authors of [36] propose a self-supervised PAD framework that utilizes self-supervision training strategy to learn local presentation attack cues from a face image. They finetune their fully convolutional network with randomly selected arbitrary-size face patches. Unlike our work, they also use partial presentation attack instruments, such as cut-out masks, therefore they assume that attack instruments have local regions over the face. [37] proposes using a human-like approach where they take a global look at first and then they use local patches for

more discriminative information. Unlike many patch-based approaches, they leverage reinforcement-learning to find face-spoofing-related information from patches of faces. Their agent tries to predict locations of patches and location of the spoofing cues. Instead of raw RGB pixels of the patches their agent works on feature maps of the patches. In [38] authors propose Destruction and Combination Network (DCN) which utilizes their novel local relation modeling, content combination, and structure destruction modules. Structure destruction module is similar to create jigsaw puzzles pretext task proposed in [39]. After creating jigsaw puzzle from a face, they combine patches of different subjects. Their experimental results also support our arguments in [40] that shuffled patches increase the generalization capabilities of the PAD models. In contrast to their work, we do not employ any similarity measurement or reflection estimation. Instead, we utilize patch-wise binary supervision directly.

3. METHODOLOGY

In the following subsections, we first explain the used face detection and face alignment methods, as well as the image partitioning scheme. Then we provide information about the employed model architecture, loss function, and details of training.

3.1 Preprocessing

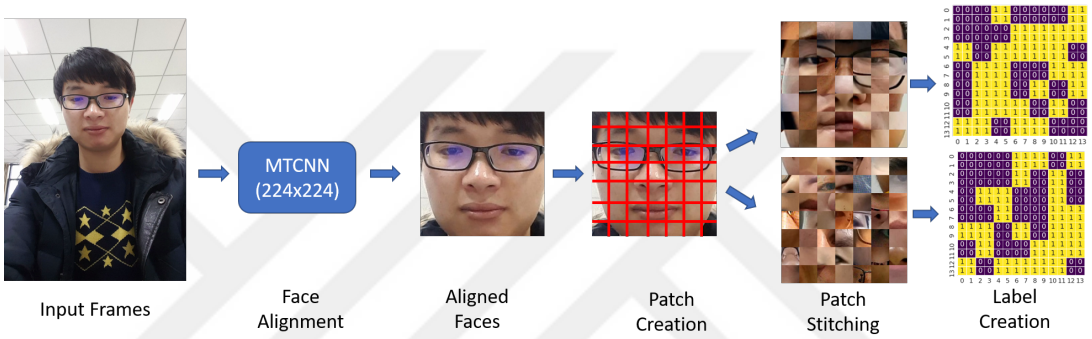


Figure 3.1 : Our overall pipeline of input and ground truth label creation starts with cropping and geometrically aligning faces. For geometric normalization, we use two eye centers. Aligned faces are divided into 32×32 pixels patches. After all patches are created for the dataset, we combine different patches of different real or spoof images. This combination process is called stitching. Stitching can be made with two different methods. The upper row shows controlled stitching, which tries to keep the facial structure intact. The bottom row shows random stitching, which takes random parts of the faces. The final step is 14×14 ground truth label creation, which contains 1 for real patches and 0 for fake patches.

It has been shown that face recognition systems can perform much better when given inputs are preprocessed to enhance the captured image [41]. Preprocessing is much more important in an uncontrolled scenario where distance between face and input sensor can vary. In our preprocessing pipeline we first detect faces and face landmarks in the given frame with MTCNN [42] face detector. Then by using Bob framework [43,44] we align the detected faces with respect to the eye coordinates and crop these aligned faces to 224×224 resolution. After that, we create patches from each face image by dividing it using a 7×7 grid. This way, we get 49 face patches of $32 \times$

32 resolution for each face image. We then combine these patches to create a new 224×224 resolution image. Our method produces 14×14 output map. Each face patch corresponds to a 2×2 location in the ground truth 14×14 label map. Bona fide patches have 1 as the label and fake patches have 0 as the label. We call this process *patch stitching*. We use two different strategies while stitching the patches. In the first approach, which we call *random stitching*, we completely randomize the patches and do not care about facial structure while combining different patches. Therefore, in this method, we get completely shuffled face images. As we randomly choose patches, the same parts of faces can be found in stitched images. For example, multiple noses or eyes can be seen in the bottom part of the Figure 3.1. In the second approach, which we call *controlled-stitching*, we combine patches of different faces while keeping the facial structure as much as possible. Therefore, we actually create an input that resembles a face and consists of multiple subjects' face parts. The input contains both bona fide and attack patches, because of that stitched images must have labels for each patch separately. In our experiments we use the former approach which gives better performance in inter-dataset experiments. The overview of the proposed method is illustrated in Figure 3.2.

3.2 Model Architecture and Training

We use a deep CNN network that takes 224×224 resolution input images and creates 14×14 output maps. We use the model that was proposed in [27] which is based on DenseNet-161 [45] architecture. However, we do not have the final linear and sigmoid layers. Our 14×14 output map selection is to be exactly comparable with our baseline [27]. The model contains the first eight layers of the DenseNet [45] and we use ImageNet [25] pretrained weights. At the end of eight layers, we add a 1×1 channel-wise convolution to produce a $14 \times 14 \times 1$ feature map which is the model output. Our proposed training method is not specific to the architecture, it can be replaced with any kind of CNN architecture. In order to show the model invariance of our training method, we also applied our method to CDCN [33], which uses Central Difference Convolution and pseudo-depth maps to detect presentation attacks. As CDCN uses different architecture, we used 256×256 input images and 32×32 depth

maps. Therefore, instead of 14×14 feature map, we use 32×32 feature maps for the CDCN. We use input images that contain multiple patches together, therefore, we do not have a binary label for each input image. In training time, we use 14×14 pixel-wise binary labels to train the model with Binary Cross Entropy (BCE) loss. We assign ground truth $y = 0$ for patches that come from attack images and $y = 1$ for patches that come from bona fide images as explained in Chapter 3.1. The equation for pixel-wise loss is given in Equation 3.1.

$$\mathscr{L}_{pixel-wiseBCE} = -(y(log(p)) + (1-y)log(1-p)) \tag{3.1}$$

In Equation 3.1 p is the 14×14 model output that contains probability values between 0 and 1. We minimize this loss with Adam [46] optimizer. We use 0.001 as the initial learning rate and halve this value at each 10th epoch. We use 32 as batch size and we generally train our methods for 45 epochs. We use horizontal flip and color jitter as data augmentation. Moreover, our patch shuffling also acts like an augmentation. In the test time, we give the aligned face image as input to the model and our model produces 14×14 output. We calculate the mean probability score by using the 14×14 output, then we use this probability as our liveness score. If the score is higher than the predefined threshold, we classify the given input as bona fide, else we classify it as an attack.

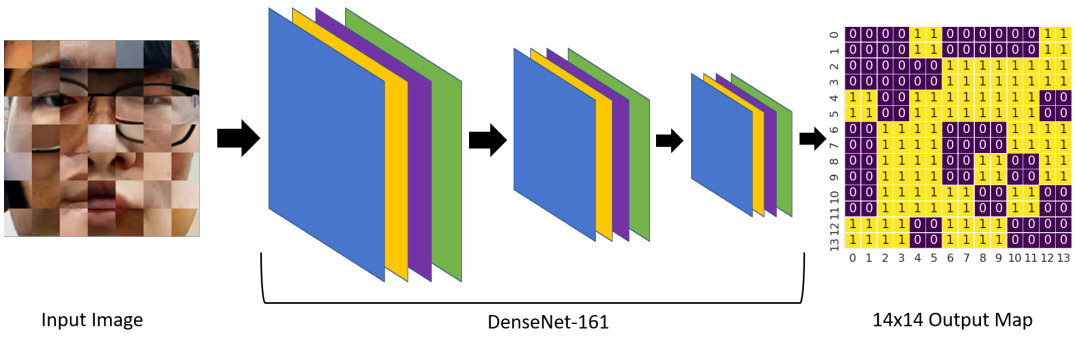


Figure 3.2 : Overview of the proposed method which uses combined patches of different faces with pixel-wise binary supervision. Input image contains 49 different patches of real or fake images. Given input is passed from DenseNet-161 architecture until it produces 14×14 output map. Ground truth map also has 14×14 binary labels.

4. DATASETS AND EXPERIMENTAL RESULTS

This section presents the datasets that have been used to evaluate the performance of the proposed approach, the experiments carried out, the obtained results, and discussions.

4.1 Datasets and Metrics

We use four different datasets for the experiments, namely, Replay Mobile [10], OULU-NPU [7], SiW [8], and Sodec Real-World dataset.

4.1.1 Replay Mobile Dataset

Replay Mobile dataset [10] consists of 1190 video clips of 40 subjects. Each subject has ten videos under different illumination and background conditions. The dataset contains paper and replay presentation attacks under five different illumination conditions. To imitate replay attack scenarios subject videos are displayed on a 1080p matte monitor, then recorded from it. For all matte screens, a stand has been used to hold capturing device still. In print attacks, the faces of the subjects are printed on A4 paper then put on a stationary surface. For print attacks, if capturing device has been held by a spoofer it is considered as the "hand" category or if it has been held by a stand then it is considered as the "fixed" category. The dataset has *"grandtest"* protocol for evaluating the global performance of an algorithm. We report our results on the *"grandtest"* protocol. We use the train, development, and test sets of the dataset that were provided with the dataset for training and evaluating our models.

4.1.2 OULU-NPU Dataset

OULU-NPU dataset [7] is one of the biggest PAD datasets with over 5900 videos of 55 subjects. Videos were recorded in three different environments. All of the videos have 1080p resolution and are taken by six different mobile phone cameras. The

dataset has both print and replay attacks with two different printers and two different electronic displays. There are four protocols for evaluating a method's generalization capability. The protocol names were set based on the level of difficulty. In all of these protocols subjects in training, development, and test sets do not overlap. Protocol 1 contains unknown background and illumination conditions in the test set to evaluate the generalization of the face PAD methods. The first two sessions are reserved for training and the last session is allocated for testing. Protocol 2 evaluates the generalization across different attack devices, such as different electronic displays and printers. Protocol 3 uses recordings from different phones to test the methods' generalizability across different input sensors. Protocol 4 evaluates the method across previously unseen backgrounds, illumination conditions, attacks, and input sensors. Therefore, it is the most challenging protocol. It combines all the restrictions in the previous protocols. Moreover, it contains less training data. We report mean and standard deviation values on Protocols 3 and 4 which use the leave-one-out strategy for six different phone cameras.

4.1.3 SiW Dataset

SiW dataset [8] is one of the largest high-quality antispoofing datasets. It has over 4400 videos of 165 subjects collected over four different sessions. All videos have 1080p resolution and contain variations of distance, pose, illumination, and expression. It contains both print and replay attacks. The dataset has three protocols to test the generalization capability of the models. We utilize this dataset for inter-dataset experiments because of the following reasons: it contains one of the highest number of subjects among PAD datasets, it contains different poses and expressions, and the acquisition device changes its distance to subjects, which is a common behavior in the real-world PAD attempts.

4.1.4 Sodec Real-World Dataset

Sodec Real-World dataset has been collected to simulate the real-world presentation attack scenarios. It contains more than 51k frames of 31 different subjects. There are 16 male and 15 female subjects. It contains only replay attacks over three

(a) Samples from OULU-NPU dataset

(b) Samples from Sodec Real-World dataset

Figure 4.1 : Sample images from OULU-NPU and Sodec Real-World datasets. In OULU-NPU dataset same input devices are employed, whereas in Sodec Real-World dataset different input devices are used. Different background, illumination, and pose variation are present in Sodec Real-World dataset.

different presentation attack instruments (PAI), namely, mobile phone, computer display, and television. Unlike controlled datasets, every subject recorded videos of themselves with their mobile phones in their own home/environment. Moreover, subjects turned 360 degrees around themselves while holding the phone to capture different backgrounds, illumination conditions, and pose variations. We show some examples from the dataset in Figure 4.1. It is easy to observe that lighting conditions and background in OULU-NPU dataset are similar. However, in the Sodec Real-World dataset, severe illumination and background differences can be seen just like in real-world scenarios. We utilize this dataset in inter-dataset experiments to test models on real-world attacks.

4.2 Experimental Results

In all of our experiments, we used ISO/IEC 30107-3 [3] standards, which are standard metrics for the PAD. Attack Presentation Classification Error Rate (APCER), Bona fide Presentation Classification Error Rate (BPCER), Half Total Error Rate (HTER) along with the Average Classification Error Rate (ACER) on the test sets are reported.

$$ACER = \frac{APCER + BPCER}{2} \tag{4.1}$$

$$HTER = \frac{FRR + FAR}{2} \tag{4.2}$$

For all of our experiments, we determine the threshold according to the equal error rate (EER) criterion that we calculate on the development set. The average of False Recognition Rate (FRR) and False Acceptance Rate (FAR) is equal to HTER. These values are calculated using the formulas given in Equation 4.1 and Equation 4.2. We use OULU-NPU and Replay Mobile datasets for our intra-dataset experiments.

Table 4.1 : Intra-dataset test results of Replay-Mobile *"grandtest"* protocol. The first column represents the Equal Error Rate (EER) in percentage and the second represents the Half Total Error Rate (HTER) in percentage.

Model	**EER (%)**	**HTER (%)**
CDCN	**0.0**	**0.0**
DeepPixBis	**0.0**	**0.0**
DeepPixBis+DPS	**0.0**	**0.0**

As explained before, we report APCER, BPCER, and ACER values obtained by the tested models. In all of our experiments we use random stitching as our patch stitching strategy. We compare our method with CDCN, CDCN++ [33,34], and DeepPixBis [27] models. CDCN++ is a state-of-the-art method that uses pseudo-depth maps and central difference convolution that specificially focuses on artefact traces. CDCN++ is the extended version of the CDCN model that utilizes neural architecture search (NAS) based backbone with multiscale attention fusion module. NAS operation specifically optimizes the model architecture for the given dataset. We integrate our method with both DeepPixBis and CDCN models. Therefore, we are using the exact same architecture when we compare the results. In the tables, we present our method under the name of DeepPixBis+DPS and CDCN+DPS, where DPS stands for deep patchwise supervision. As we use DeepPixBis and CDCN architectures as they were proposed, our results are directly comparable in terms of model capacity and computing time. For intra-dataset experiments we also present the results of different state-of-the-art methods such as DC-CDN [35], CDCN-PS [28], DCN [38]. Table 4.1 presents the obtained results on the Replay Mobile dataset, *"grandtest"* protocol. Similar to our method, in the Replay-Mobile dataset, most of the recent methods report 0% error. In Table 4.2, we report the intra-dataset results on the OULU-NPU dataset. Table 4.2 shows that our method outperforms DeepPixBis and CDCN on Protocol 3 and Protocol 4, which are the hardest protocols of the dataset. Integrating our training strategy into both of these networks provides substantial improvement. Moreover, applying our method on CDCN backbone performs better than all the state-of-the-art methods on Protocol 2, 3, and 4 in terms of ACER metric. These protocols have smaller training data and their test data is not very similar to the training data in terms of environment, PAI, and PA acquisition device. Our method has a higher error rate on Protocol 1, but we believe that our method is well suited for generalization whereas other methods gain an advantage of similar training and testing sets in these protocols.

Table 4.2 : Intra-dataset test results on OULU-NPU dataset.

Pr	Model	APCER (%)	BPCER (%)	ACER (%)
1	CDCN	**0.4**	1.7	1.0
	CDCN++	**0.4**	**0.0**	**0.2**
	DC-CDN	0.5	0.3	0.4
	CDCN-PS	**0.4**	1.2	0.8
	DCN	1.3	0.0	0.6
	LMFD-PAD	1.4	1.6	1.5
	DeepPixBis	0.8	**0.0**	0.42
	DeepPixBis+DPS	2.1	2.1	2.1
	CDCN+DPS	0.95	2.1	1.1
2	CDCN	1.5	1.4	1.5
	CDCN++	1.8	0.8	1.3
	DC-CDN	**0.7**	1.9	1.3
	CDCN-PS	1.4	1.4	1.4
	DCN	2.2	2.2	2.2
	LMFD-PAD	3.1	0.8	2.0
	DeepPixBis	11.3	**0.5**	5.9
	DeepPixBis+DPS	6.2	6.2	6.2
	CDCN+DPS	1.3	1.2	**1.2**
3	CDCN	2.4 ± 1.3	2.2 ± 2.0	2.3 ± 1.4
	CDCN++	1.7 ± 1.5	2.0 ± 1.2	1.8 ± 0.7
	DC-CDN	2.2 ± 2.8	1.6 ± 2.1	1.9 ± 1.1
	CDCN-PS	1.9 ± 1.7	2.0 ± 1.8	2.0 ± 1.7
	DCN	2.3 ± 2.7	$\mathbf{1.4 \pm 2.6}$	1.9 ± 1.6
	LMFD-PAD	3.5 ± 3.2	3.3 ± 3.2	3.4 ± 3.1
	DeepPixBis	11.6 ± 19.5	10.5 ± 14.0	11.1 ± 9.4
	DeepPixBis + DPS	6.1 ± 2.5	6.3 ± 2.5	6.2 ± 2.5
	CDCN+DPS	$\mathbf{1.5 \pm 2.1}$	1.6 ± 1.5	$\mathbf{1.55 \pm 1.8}$
4	CDCN	4.6 ± 4.6	9.2 ± 8.0	6.9 ± 2.9
	CDCN++	$\mathbf{4.2 \pm 3.4}$	5.8 ± 4.9	5.0 ± 2.9
	DC-CDN	5.4 ± 3.3	2.5 ± 4.2	4.0 ± 3.1
	CDCN-PS	2.9 ± 4.0	5.8 ± 4.9	4.8 ± 1.8
	DCN	6.7 ± 6.8	$\mathbf{0.0 \pm 0.0}$	$\mathbf{3.3 \pm 3.4}$
	LMFD-PAD	4.5 ± 5.3	2.5 ± 4.1	$\mathbf{3.3 \pm 3.1}$
	DeepPixBis	36.6 ± 29.6	13.3 ± 16.7	25.0 ± 12.0
	DeepPixBis + DPS	11.5 ± 7.8	11.5 ± 7.8	11.5 ± 7.8
	CDCN+DPS	$\mathbf{2.8 \pm 3.7}$	3.9 ± 4.2	$\mathbf{3.3 \pm 3.9}$

4.3 Real-World Experiments

There are many face antispoofing datasets that are collected in controlled environments. Most of the datasets only consider two or three backgrounds with controlled illumination changes. However, in real-world applications, there are various backgrounds, illumination, poses, and expression changes. Moreover, attackers are more careful when creating a face presentation attack. We collected a dataset to simulate the real-world use case of presentation attacks. We trained CDCN [33,34], DeepPixBis [27], and our model on the SiW dataset Protocol 1. In these experiments, we train CDCN only, because CDCN++ is a product of neural architecture search on a dataset. Therefore it is an optimized version of the CDCN model for the specific dataset. Moreover, we also integrated our proposed training method with CDCN and DeepPixBis to further show that our method is compatible with different architectures and increases the performance of these methods. For these experiments we chose SiW dataset because it is one of the most representative datasets in terms of distance, pose, illumination, and expression changes among PAD datasets. From Table 4.3 we can see that state-of-the art CDCN model has achieved the lowest error rate on the SiW dataset, but it obtains a higher error rate on the Sodec Real-World Dataset. Results indicate that when our method is integrated with different baselines, our models are able to outperform DeepPixBis and CDCN without modifying their neural network

Table 4.3 : Experiment results of SiW (Protocol 1) and inter-dataset test results on Sodec Real-World Dataset. Reported metrics in the table represents the ACER values in percentage (%)

	Trained on SiW	
Model	tested on SiW	tested on Real-World Dataset
CDCN	<u>0.12</u>	12.52
LMFD-PAD	1.32	7.25
DeepPixBis	3.68	12.05
DeepPixBis+DPS	2.15	<u>5.24</u>
CDCN+DPS	**0.11**	**4.76**

architecture on inter-dataset and intra-dataset experiments of both the SiW dataset and Sodec Real-World Dataset. The results show that our proposed method is useful

and more suitable for real-world inter-dataset scenarios, which is the hardest task to perform.

5. ABLATION STUDY

In this section we investigate the effects of hyperparameters on the performance of the method. We present our results on OULU-NPU dataset.

5.1 Patch Size

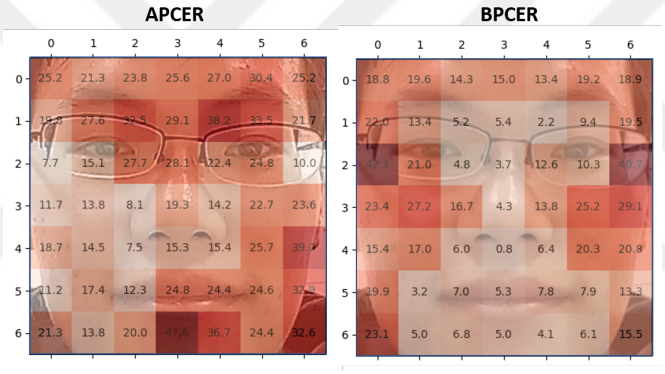
In our ablation, we first investigate the proposed method's behavior for different patch sizes. Our training method requires face patches of different bona fide and attack images. We show that using stitched patches with patch-wise binary cross-entropy loss increases the generalization power of the model. Yet, patch size plays an important role on the performance. We can treat the whole image as one patch in an extreme case using no stitching or each pixel as one patch. In our ablation study, we use three different patch sizes to assess how patch size affects the performance. In Table 5.1, we present the proposed method's performance when we use 16×16, 32×32, and 64×64 patch sizes. Our ablation study shows that 32×32 pixels resolution patch size provides the best performance in most of the protocols. Therefore, in our experiments in Chapter 4 we used 32×32 patch size.

Table 5.1 : Performance of our method for different patch sizes.

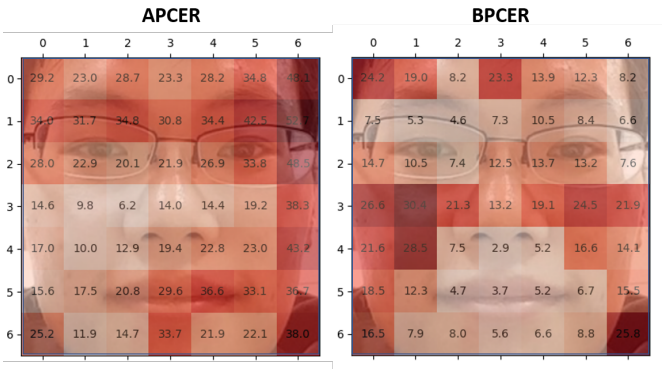
Prot.	Patch Size (px)	APCER (%)	BPCER (%)	ACER (%)
1	16x16	3.05	3.42	2.23
	32x32	**2.14**	**2.14**	**2.14**
	64x64	3.7	3.02	3.36
2	16x16	6.86	**6.02**	6.44
	32x32	6.22	6.26	6.26
	64x64	**5.92**	6.11	**6.09**
3	16x16	6.45 ± 1.35	6.21 ± 3.15	6.33 ± 2.25
	32x32	$\mathbf{6.20 \pm 2.57}$	$\mathbf{6.20 \pm 2.55}$	$\mathbf{6.20 \pm 2.56}$
	64x64	7.83 ± 3	8.03 ± 3.55	7.93 ± 3.27
4	16x16	11.85 ± 5.16	11.89 ± 8.84	11.87 ± 7
	32x32	$\mathbf{11.51 \pm 7.86}$	$\mathbf{11.58 \pm 7.84}$	$\mathbf{11.54 \pm 7.84}$
	64x64	17.42 ± 6.84	18.24 ± 7.02	17.83 ± 6.93

5.2 Patch Location

In presentation attacks, some parts of the images can be more informative to detect spoof traces. Also, models may focus on information from unrelated parts of the face, such as from background in the case of an overfit to the dataset. In our approach, we average the 14×14 output score map, which shows the model's decision for each patch. Therefore, we are actually using the average liveness score of all patches. However, if some patch locations perform significantly worse than others, it might negatively affect the final decision. In order to investigate our model's decisions, we calculated APCER (%) and BPCER (%) values according to each patch location. So, we treated each patch output as a different binary classifier. As we explained in Chapter 4, our model produces 2×2 output for each patch location. Therefore, for each patch, we produce four scores. For this ablation study, we use the average of the four



(a) OULU-NPU Protocol 1



(b) OULU-NPU Protocol 2

Figure 5.1 : APCER(%) and BPCER(%) according to the decision of each patch solely. Given values are the average patch performance over whole test dataset. Illustrated face helps readers to better understand patch locations over an average human face.

values to decide the score of each patch. This calculation gives us a 7×7 output map, where each location corresponds to a patch. In Figure 5.1, we see that outer patches affect the performance significantly worse than inner parts of the face. This might be due to the fact that most of outer patches mostly correspond to the background. From this observation, we can conclude that the nose, mouth, and eye region of the face are more suitable to detect presentation attacks.

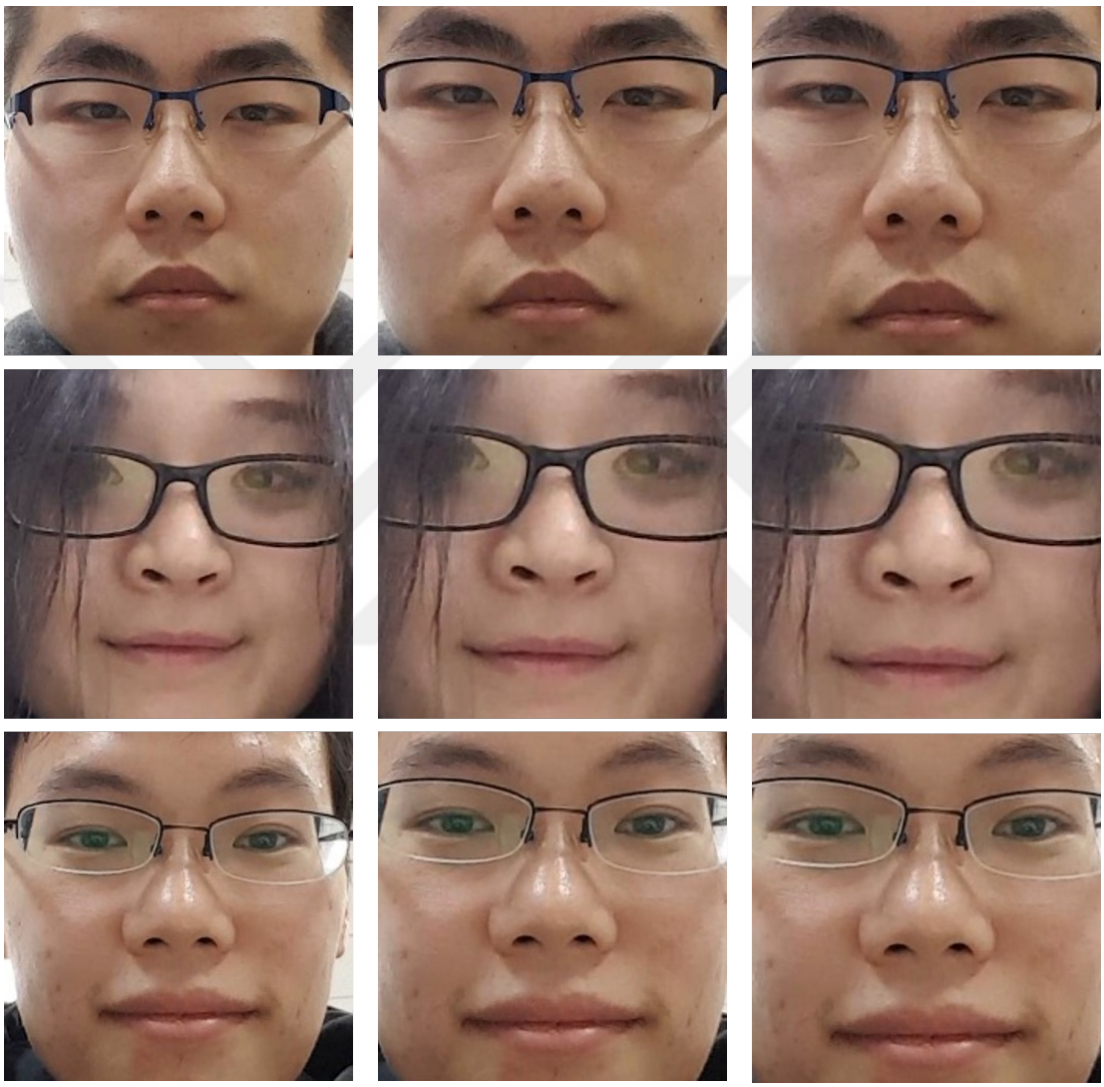


Figure 5.2 : Example Tight (first column), Tighter (second column), and Tightest (last column) face crops from OULU-NPU dataset. Distance between left and right eye centers differ for different crops. Tight, tighter and tightest refer to crops with 112, 148, and 166 pixels eye center distance, respectively.

5.3 Facial Bounding Box Size

We also experiment on the effect of facial bounding box size. Previous works [47]–[49] showed that how the cropping of face images affect face recognition and PAD performance directly. Therefore, we investigate the effect by using three different facial bounding box sizes to crop the face images. We increase the distance between eye centers to get tighter bounding boxes as we geometrically normalize (align) our faces with respect to the eye centers. We used 112 pixels eye center distance for our experiments in Chapter 4. We increased this distance by 36 and 54 pixels in order to get tighter and tightest crops, respectively. Figure 5.2 contains sample face images aligned with different inter-ocular distances. It can be seen from Table 5.2 that Tight or Tighter face crops perform better than Tightest face crops. This is because of in Tightest crop some facial regions such as chin and mouth can stay outside of the face crop in some subject images as we use geometrically normalized crops according to the eyes. Even though performances are relatively close to each other in all protocols, Tight crops give the best results for every protocol except protocol four. In Tight crops, some background information is present at the expense of covering the entire face. In the Tighter and Tightest crops, background information is completely removed, however, this also led to the removal of some facial parts too, for example, a part of the forehead.

Table 5.2 : Effect of different bounding box sizes on test results of OULU-NPU dataset. Tight, tighter and tightest refer to crops with 112, 148, and 166 pixels eye center distance, respectively.

Prot.	Bounding Box	APCER(%)	BPCER (%)	ACER (%)
1	Tight	**2.14**	**2.14**	**2.14**
	Tighter	2.26	2.18	2.22
	Tightest	2.28	2.26	2.27
2	Tight	**6.22**	6.26	**6.24**
	Tighter	6.25	**6.24**	**6.24**
	Tightest	6.34	6.30	6.32
3	Tight	**6.10 ± 2.57**	6.30 ± 2.55	**6.20 ± 2.55**
	Tighter	6.32 ± 2.20	**6.28 ± 2.13**	6.30 ± 2.16
	Tightest	6.42 ± 1.87	6.40 ± 1.96	6.41 ± 1.91
4	Tight	11.51 ± 7.86	11.58 ± 7.84	11.54 ± 7.84
	Tighter	**11.49 ± 6.14**	**11.40 ± 6.22**	**11.44 ± 5.18**
	Tightest	12.11 ± 5.12	12.09 ± 6.10	12.10 ± 5.61

Slightly better performance on Tight crops could be the result of some spoof traces like paper wrinkles and screen reflections that can be detected by using the outer parts of the crops.

5.4 Stitching Type

Our last experiment for the ablation study is the effect of different stitching types. As we explained in Chapter 3.1, our input images are made of multiple patches of different people. While we combine different patches, which we call patch stitching, we use two different strategies. The first one is the random stitching, and the second one is controlled stitching. Details of these strategies are given in Chapter 3.1. In Table 5.3 we present our experimental results of using these stitching strategies. From the Table 5.3, we see that random stitching consistently outperforms the controlled stitching in all protocols of the OULU-NPU dataset. However, the performances of both strategies are close to each other. It can also be seen that the performance difference is larger in protocols 3 and 4. Better results on the random stitching strategy can be explained by the fact that the generalization of the model increases as the number of possible patch combinations increases. This is expected to reduce the bias on specific facial structures within the dataset.

Table 5.3 : Effect of different stitching methods on OULU-NPU dataset. Random stitching refers to combining real and fake patches of different people randomly, whereas controlled stitching refers to combining the patches while trying to keep the facial structure intact. Therefore in controlled stitching, stitched images resemble a face which consists of multiple different subjects.

Prot.	Stitching Strategy	APCER(%)	BPCER (%)	ACER (%)
1	Random	**2.14**	**2.14**	**2.14**
	Controlled	2.33	2.40	2.36
2	Random	**6.22**	6.26	**6.24**
	Controlled	6.41	6.38	6.39
3	Random	**6.10 ± 2.57**	**6.30 ± 2.55**	**6.20 ± 2.55**
	Controlled	6.70 ± 1.15	6.88 ± 1.32	6.79 ± 1.23
4	Random	**11.51 ± 7.86**	**11.58 ± 7.84**	**11.54 ± 7.84**
	Controlled	12.10 ± 4.25	12.02 ± 4.76	12.06 ± 4.50

5.5 Discussion

Experimental results and ablation study indicate that the proposed training method can improve PAD performance. More specifically, we see a substantial performance increase in terms of generalization capability. Our experiments in Chapter 4.3 show that the proposed method is useful for real-world scenarios where illumination and background conditions vary significantly. Our ablation study provides in depth insight for our model. Results show that using tiny patches degrades the performance. Moreover, inner face regions such as the nose, mouth, and eyes are the most critical parts to detect presentation attacks for our model. Results also indicate that tight crops are helpful for our method until it starts to leave out face regions, such as the chin and mouth. Table 5.1 shows that limiting background information more might increase the performance slightly. However, taking much smaller face crops like in the *tightest crop* case might degrade the performance since the amount of facial information also decreases. Furthermore, some presentation attack traces like screen reflection or paper crease generally occur outside the face area. Therefore, using the Tightest bounding box sometimes degrades the performance.

6. CONCLUSIONS

In this thesis, we proposed a new training method for deep learning-based PAD models by utilizing different face patches of live and spoof images, instead of a single face image, in training time. Our proposed method, namely Deep Patch-Wise Supervision for Presentation Attack Detection (DPS-PAD), is able to produce more samples for training by combining different patches, therefore, it is effective to use with smaller PAD datasets. We show that training models with pixel-wise binary loss and shuffled face patches can improve PAD performance.

We extensively tested our proposed method to demonstrate the effectiveness of deep patch-wise supervision. We also profoundly analyzed our method's components by investigating different patch sizes, the effect of patch locations, and face crop size. Our proposed method improves both CDCN and DeepPixBis performances on OULU-NPU Protocol 3 and 4, which are the most challenging protocols of the dataset. Furthermore, when we apply our method on CDCN, we get superior results than state-of-the-art PAD methods on OULU-NPU Protocol 2,3, and 4. Moreover, the proposed method performs much better when we compare the models with state-of-the-art PAD methods on SiW and Sodec Real World Dataset.

We plan to extend the Sodec Real World dataset to print attacks as future work. Furthermore, we are planning to modify the backbone architecture of the model and analyze the effects of different patch creation methods on model behavior. Created patch stitching method can be applied to depth-based state-of-the-art algorithms to improve their performances even further.

REFERENCES

[1] **Hernandez-Ortega, J.**, **Fierrez, J.**, **Morales, A. and Galbally, J.**, (2019). Introduction to face presentation attack detection, Handbook of Biometric Anti-Spoofing, Springer, pp.187–206.

[2] **Jia, S.**, **Guo, G.**, **Xu, Z. and Wang, Q.** (2020). Face presentation attack detection in mobile scenarios: A comprehensive evaluation, *Image and Vision Computing*, *93*, 103826.

[3] **ISO/IEC JTC1 SC37** (2016). Information Technology - Biometric presentation attack detection - Part 3: Testingand Reporting, **Technical Report**, International Organization for Standardization, Geneva, CH.

[4] **Czajka, A. and Bowyer, K.W.** (2018). Presentation attack detection for iris recognition: An assessment of the state-of-the-art, *ACM Computing Surveys (CSUR)*, *51*(4), 1–35.

[5] **González-Soler, L.J.**, **Gomez-Barrero, M.**, **Chang, L.**, **Pérez-Suárez, A. and Busch, C.** (2021). Fingerprint Presentation Attack Detection Based on Local Features Encoding for Unknown Attacks, *IEEE Access*, *9*, 5806–5820.

[6] **Tolosana, R.**, **Vera-Rodriguez, R.**, **Fierrez, J. and Ortega-Garcia, J.**, (2019). Presentation attacks in signature biometrics: types and introduction to attack detection, Handbook of Biometric Anti-Spoofing, Springer, pp.439–453.

[7] **Boulkenafet, Z.**, **Komulainen, J.**, **Li, L.**, **Feng, X. and Hadid, A.** (2017). OULU-NPU: A mobile face presentation attack database with real-world variations, *IEEE Intl. Conference on Automatic Face and Gesture Recognition*.

[8] **Liu, Y.**, **Jourabloo, A. and Liu, X.** (2018). Learning Deep Models for Face Anti-Spoofing: Binary or Auxiliary Supervision, *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.389–398.

[9] **Zhang, S.**, **Liu, A.**, **Wan, J.**, **L., Y.**, **Guo, G.**, **Escalera, S.**, **Escalante, H.J. and Li, S.Z.** (2020). CASIA-SURF: A Large-Scale Multi-Modal Benchmark for Face Anti-Spoofing, *IEEE Transactions on Biometrics, Behavior, and Identity Science*, *2*(2), 182–193.

[10] **Costa-Pazo, A.**, **Bhattacharjee, S.**, **Vazquez-Fernandez, E. and Marcel, S.** (2016). The REPLAY-MOBILE Face Presentation-Attack Database, *Proceedings of the Intl. Conference on Biometrics Special Interests Group (BIOSIG)*, pp.1–7.

[11] **Liu, Y.**, **Stehouwer, J.**, **Jourabloo, A. and Liu, X.** (2019). Deep tree learning for zero-shot face anti-spoofing, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.4680–4689.

[12] **Heusch, G.**, **George, A.**, **Geissbühler, D.**, **Mostaani, Z. and Marcel, S.** (2020). Deep Models and Shortwave Infrared Information to Detect Face Presentation Attacks, *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 399–409.

[13] **Li, X.**, **K., J.**, **Z., G.**, **Y., P.C. and Pietikainen, M.** (2016). Generalized face anti-spoofing by detecting pulse from face videos, *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp.4244–4249.

[14] **Maatta, J.**, **Hadid, A. and Pietikainen, M.** (2011). Face spoofing detection from single images using micro-texture analysis, *2011 Intl. Joint Conference on Biometrics (IJCB)*, pp.1–7.

[15] **Atoum, Y.**, **Liu, Y.**, **Jourabloo, A. and Liu, X.** (2017). Face anti-spoofing using patch and depth-based CNNs, *IEEE Intl. Joint Conference on Biometrics (IJCB)*, pp.319–328.

[16] **Xu, Y.**, **Wang, Z.**, **Han, H.**, **Wu, L. and Liu, Y.** (2021). Exploiting Non-uniform Inherent Cues to Improve Presentation Attack Detection, *IEEE International Joint Conference on Biometrics (IJCB)*, pp.1–8.

[17] **Yang, X.**, **Luo, W.**, **Bao, L.**, **Gao, Y.**, **Gong, D.**, **Zheng, S.**, **Li, Z. and Liu, W.** (2019). Face anti-spoofing: Model matters, so does data, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.3507–3516.

[18] **Wang, Z.**, **Yu, Z.**, **Zhao, C.**, **Zhu, X.**, **Qin, Y.**, **Zhou, Q.**, **Zhou, F. and Lei, Z.** (2020). Deep spatial gradient and temporal depth learning for face anti-spoofing, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.5042–5051.

[19] **Ming, Z.**, **Visani, M.**, **Luqman, M.M. and Burie, J.** (2020). A Survey On Anti-Spoofing Methods For Face Recognition with RGB Cameras of Generic Consumer Devices, *arxiv.org/abs/2010.04145*.

[20] **Li, X.**, **Wan, J.**, **Jin, Y.**, **Liu, A.**, **Guo, G. and Li, S.Z.** (2020). 3DPC-net: 3D point cloud network for face anti-spoofing, *IEEE International Joint Conference on Biometrics (IJCB)*, IEEE, pp.1–8.

[21] **Wang, Z.**, **Xu, Y.**, **Wu, L.**, **Han, H.**, **Ma, Y. and Ma, G.** (2021). Multi-Perspective Features Learning for Face Anti-Spoofing, *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp.4116–4122.

[22] **Wang, C. and Zhou, J.** (2022). An adaptive index smoothing loss for face anti-spoofing, *Pattern Recognition Letters*, *153*, 168–175.

[23] **Guo, J.**, **Zhu, X.**, **Xiao, J.**, **Lei, Z.**, **Wan, G. and Li, S.Z.** (2019). Improving face anti-spoofing by 3d virtual synthesis, *2019 International Conference on Biometrics (ICB)*, IEEE, pp.1–8.

[24] **Yang, J.**, **Lei, Z. and Li, S.Z.** (2014). Learn Convolutional Neural Network for Face Anti-Spoofing, *arxiv.org/abs/1408.5601*.

[25] **Krizhevsky, A.**, **Sutskever, I. and Hinton, G.** (2012). ImageNet Classification with Deep Convolutional Neural Networks, *Advances in Neural Information Processing Systems*, volume 25.

[26] **Yu, Z.**, **Qin, Y.**, **Li, X.**, **Zhao, C.**, **Lei, Z. and Zhao, G.** (2021). Deep learning for face anti-spoofing: A survey, *arXiv preprint arXiv:2106.14948*.

[27] **George, A. and Marcel, S.** (2019). Deep Pixel-wise Binary Supervision for Face Presentation Attack Detection, *12th IAPR Intl. Conference on Biometrics (ICB)*, pp.1–8.

[28] **Yu, Z.**, **Li, X.**, **Shi, J.**, **Xia, Z. and Zhao, G.** (2021). Revisiting Pixel-Wise Supervision for Face Anti-Spoofing, *IEEE Transactions on Biometrics, Behavior, and Identity Science*, *3*(3), 285–295.

[29] **Adelson, E.H.**, **Anderson, C.H.**, **Bergen, J.R.**, **Burt, P.J. and Ogden, J.M.** (1984). Pyramid methods in image processing, *RCA engineer*, *29*(6), 33–41.

[30] **Qin, Y.**, **Yu, Z.**, **Yan, L.**, **Wang, Z.**, **Zhao, C. and Lei, Z.** (2021). Meta-teacher for Face Anti-Spoofing, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–1.

[31] **Qin, Y.**, **Zhao, C.**, **Zhu, X.**, **Wang, Z.**, **Yu, Z.**, **Fu, T.**, **Zhou, F.**, **Shi, J. and Lei, Z.** (2020). Learning meta model for zero-and few-shot face anti-spoofing, *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp.11916–11923.

[32] **Kang, G.**, **Dong, X.**, **Zheng, L. and Yang, Y.** (2017). Patchshuffle regularization, *arXiv preprint arXiv:1707.07103*.

[33] **Yu, Z.**, **Zhao, C.**, **Wang, Z.**, **Qin, Y.**, **Su, Z.**, **Li, X.**, **Zhou, F. and Zhao, G.** (2020). Searching Central Difference Convolutional Networks for Face Anti-Spoofing, *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.5294–5304.

[34] **Yu, Z.**, **Wan, J.**, **Qin, Y.**, **Li, X.**, **Li, S.Z. and Zhao, G.** (2020). NAS-FAS: Static-dynamic central difference network search for face anti-spoofing, *IEEE transactions on pattern analysis and machine intelligence*, *43*(9), 3005–3023.

[35] **Yu, Z.**, **Qin, Y.**, **Zhao, H.**, **Li, X. and Zhao, G.** (2021). Dual-cross central difference network for face anti-spoofing, *arXiv preprint arXiv:2105.01290*.

[36] **Deb, D. and Jain, A.K.** (2020). Look locally infer globally: a generalizable face anti-spoofing approach, *IEEE Transactions on Information Forensics and Security*, *16*, 1143–1157.

[37] **Cai, R.**, **Li, H.**, **Wang, S.**, **Chen, C. and Kot, A.C.** (2021). DRL-FAS: A Novel Framework Based on Deep Reinforcement Learning for Face Anti-Spoofing, *IEEE Transactions on Information Forensics and Security*, *16*, 937–951.

[38] **Zhang, K.Y.**, **Yao, T.**, **Zhang, J.**, **Liu, S.**, **Yin, B.**, **Ding, S. and Li, J.** (2021). Structure Destruction and Content Combination for Face Anti-Spoofing, *IEEE International Joint Conference on Biometrics (IJCB)*, pp.1–6.

[39] **Noroozi, M. and Favaro, P.** (2016). Unsupervised learning of visual representations by solving jigsaw puzzles, *European conference on computer vision (ECCV)*, Springer, pp.69–84.

[40] **Kantarcı, A.**, **Dertli, H. and Ekenel, H.K.** (2021). Shuffled Patch-Wise Supervision for Presentation Attack Detection, *2021 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp.1–5.

[41] **Dharavath, K.**, **Talukdar, F.A. and Laskar, R.H.** (2014). Improving face recognition rate with image preprocessing, *Indian Journal of Science and Technology*, *7*(8), 1170–1175.

[42] **Zhang, K.**, **Zhang, Z.**, **Li, Z. and Qiao, Y.** (2016). Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks, *IEEE Signal Processing Letters*, *23*(10), 1499–1503.

[43] **Anjos, A.**, **Shafey, L.E.**, **Wallace, R.**, **Günther, M.**, **McCool, C. and Marcel, S.** (2012). Bob: a free signal processing and machine learning toolbox for researchers, *20th ACM Conference on Multimedia Systems (ACMMM), Nara, Japan*.

[44] **Anjos, A.**, **Günther, M.**, **de Freitas Pereira, T.**, **Korshunov, P.**, **Mohammadi, A. and Marcel, S.** (2017). Continuously Reproducing Toolchains in Pattern Recognition and Machine Learning Experiments, *Intl. Conference on Machine Learning (ICML)*.

[45] **Huang, G.**, **Liu, Z.**, **Van Der Maaten, L. and Weinberger, K.Q.** (2017). Densely Connected Convolutional Networks, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.2261–2269.

[46] **Kingma, D.P. and Ba, J.** (2015). Adam: A Method for Stochastic Optimization, *3rd Intl. Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[47] **Wang, K.**, **Peng, X.**, **Yang, J.**, **Meng, D. and Qiao, Y.** (2020). Region attention networks for pose and occlusion robust facial expression recognition, *IEEE Transactions on Image Processing*, *29*, 4057–4069.

[48] **Neto, P.C.**, **Sequeira, A.F. and Cardoso, J.S.** (2022). Myope Models - Are Face Presentation Attack Detection Models Short-Sighted?, *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, pp.390–399.

[49] **Mehdipour Ghazi, M. and Kemal Ekenel, H.** (2016). A comprehensive analysis of deep learning based representation for face recognition, *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp.34–41.

CURRICULUM VITAE

Name SURNAME: Alperen KANTARCI

EDUCATION:

- **B.Sc.:** 2019, Istanbul Technical University, Faculty of Computer and Informatics, Department of Computer Engineering
- **M.Sc.:** 2022, Istanbul Technical University, Graduate School, Department of Computer Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2021-2022 Istanbul Technical University Research Assistant

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **A. Kantarcı**, H. Dertli and H. K. Ekenel, "Shuffled Patch-Wise Supervision for Presentation Attack Detection," 2021 International Conference of the Biometrics Special Interest Group (BIOSIG), 2021, pp. 1-5, doi: 10.1109/BIOSIG52210.2021.9548317.

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- **A. Kantarcı**, H. Dertli, H.K. Ekenel, "Shuffled Patch-wise Supervision for Presentation Attack Detection", IEEE 20th International Conference of the Biometrics Special Interest Group (BIOSIG), Darmstadt, Germany, September 2021

- Purnapatra et. al. Face Liveness Detection Competition (LivDet-Face)-2021, IEEE International Joint Conference on Biometrics (IJCB), 2021

- F. V. Massoli, F. Falchi, **A. Kantarcı**, Ş. Aktı, H. K. Ekenel and G. Amato, "MOCCA: Multilayer One-Class Classification for Anomaly Detection," in IEEE Transactions on Neural Networks and Learning Systems, doi: 10.1109/TNNLS.2021.3130074.

- D. Engin*, **A. Kantarcı***, S. Arslan, H. K. Ekenel, Offline Signature Verification on Real-World Documents, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2020

- G. Akyol, **A. Kantarcı**, A. E. Çelik, A. C. Ak, "Deep Learning Based, Real-Time Object Detection for Autonomous Driving", IEEE 27th Signal Processing and Communications Applications Conference (SIU) 2020, Gaziantep, Turkey, April 2020

- **A. Kantarcı**, H.K. Ekenel, "Thermal to Visible Face Recognition Using Deep Autoencoders", IEEE 18th International Conference of the Biometrics Special Interest Group (BIOSIG), Darmstadt, Germany, September 2019

- R. Bulut, **A. Kantarcı**, S. Keskin, Ş. Bahtiyar, "Blockchain-Based Electronic Voting System for Elections in Turkey", IEEE 4th International Conference on Computer Science and Engineering 2019, Samsun, Turkey, September 2019