Universiteit Leiden

Master Computer Science

Text guided face forgery detection

Name:	Dimitrios Ieronymakis
Student ID:	s3372804
Date:	11/07/2023
Specialisation:	Bioinformatics
1st supervisor:	Michael Lew
2nd supervisor:	Erwin Bakker

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

1 Abstract

In recent years, technological advancements have pushed generative neural networks to unprecedented capabilities. Technologies capable of generating deepfakes and media content have emerged, facilitating the creation of hyper-realistic manipulated images, videos, and audio, to such an extent that they can deceive even the most discerning audience. Currently, the proliferation of face forgery synthesis methods has allowed individuals to exploit them for malicious purposes, including defamation and social disruption. The forensics community has been actively trying to develop methods to counter synthetically generated media, especially for face forgery detection. However, the key limitation of this task is the difficulty to achieve robust generalization performance beyond the training data. In an attempt to make a contribution to the field, we investigate the generalization capabilities of CLIP, a language-vision pre-trained model, for the task of deepfake detection, as it has not been researched previously. Through our exploration, we discover that CLIP already possesses inherent knowledge regarding deepfakes. However, we acknowledge that its performance is inadequate for the task when compared to the current State-Of-The-Art (SOTA) methods in the field. To address this, we propose a fine-tuning method, tailored for zero-shot language-vision models such as CLIP, on datasets with a limited number of classes. This method proves to be effective in enhancing the overall performance of the model and attaining SOTA generalization results on widely used deepfake detection benchmarking datasets, that are prevalent in the literature. In addition, the advantages and limitations of the proposed method are examined, providing recommendations and procedures to further enhance robustness and overall performance. By doing so, we hope to contribute to the ongoing advancements in deepfake detection, facilitating the development of more effective and reliable methods in this crucial domain.

Contents

List of Figures

List of Tables

2 Introduction

The power of generative models and their integration into big commercial products has greatly increased in recent years, making the technology extremely accessible to anyone, leading to an abuse of synthetic media, specifically deepfakes. Artificially generated content has revolutionized the way we create and consume media, but also presents significant ethical and security concerns, as it has the potential to manipulate and distort reality. This issue has been addressed multiple times in works such as [Wojewidka, 2020, Wæver and Buzan, 2020, Hancock and Bailenson, 2021]. With deepfakes becoming increasingly sophisticated, it is critical to develop effective methods for detecting artificially generated images and videos.

At first glance, face forgery detection can appear as a simple classification problem that can be solved with traditional computer vision techniques. That is only partly true. The biggest limitation these approaches face is their generalization capability, usually performing extremely well for data of the same distribution as that trained upon, yet very poorly when faced with samples of different synthesis methods. The forensics community has been actively trying to find countermeasures to the generalization problem. While early work depended heavily on detecting the discrepancies in consecutive video frames, such as [Li et al., 2018, Yang et al., 2019], attention also shifted to detecting deepfakes from single images. Some examples of this line of work are [Cao et al., 2022, Kim et al., 2021, Sun et al., 2022]. Of course, however, multi-frame-based approaches are still extremely popular, with [Guan et al., 2022] being one of the identified State-Of-The-Art (SOTA) methods in this line of work.

To the best of our knowledge however, none of the methods for deepfake detection currently available has tried to exploit the semantic understanding and generalization capabilities possible through natural language processing (NLP). Recent breakthroughs and advancements of language modeling show that when these networks are paired with substantial computational power and extensive text corpora, they can reach impressive generalization capabilities in various tasks, without being explicitly trained on any of them. In addition, following the recent advances and success of multi-modal architec-

tures, the main question we are trying to answer in this research is if the field of deepfake detection can exploit the conceptual nature of language learned by language-vision models, to improve generalization capabilities.

The proposed method in this work is based on the joint image and text representation learning paradigm introduced by [Radford et al., 2021], called Contrastive Language Image Pre-training (CLIP). The core concept of this method is to train an image and a text encoder simultaneously to produce similar embeddings, by learning to predict the correct image-description pairs in a batch. Since our problem setting is binary, we propose a novel adaptation to the original pre-training method to fine-tune CLIP on downstream tasks, where the number of classes is limited.

To understand the effects of text in the classification task we carry out various experiments, suggesting that taking advantage of the semantic understanding of text and generalization of pre-trained language-vision models, can be beneficial for deepfake detection. We show that it is possible for these kinds of architectures to reach SOTA generalization performance without many of the hustles other vision-only methods may rely on, such as heavy preprocessing of the data or complex losses to capture inconsistencies in images or sequential frames. In addition, we perform different studies that show that the original CLIP model is not robust enough to perform well on the task of deepfake detection and that a proper fine-tuning method is required to improve performance. Finally, an extensive discussion is made on the collected results and on possible improvements to the current approach.

This work aims to explore neural network generalization capabilities on out-of-distribution (OOD) data while contributing to the advancement of synthetic media detection, making safety-critical applications more robust to possible novel media manipulation techniques.

The outline of this document is as follows. The next Section 3, introduces the related work and the current SOTA approaches for the task of face forgery detection, highlighting strengths and weaknesses of each method. In Section 4 we present the fundamental work related to our method. That includes a subsection dedicated to the Transformer network, as it is the basis

for our approach's text and image encoders, and a sub-section analyzing the advancements of language-vision models. In Section 5 we introduce the datasets used for our experiments, which are some of the most commonly found in the literature. Section 6 describes the CLIP method and how it is utilized for the task of deepfake detection, while in Section 7 we describe the experimental setup. In Section 8 the experiments and results are presented. This Section is divided into multiple sub-sections, each exploring a specific aspect of the method. First, we experiment with the original model to get a grasp of the *default* capabilities and limitations of CLIP. Then we try to extract as much performance as possible to compare it with other SOTA methods. Finally, we further explore the behavior of our text embeddings to see if any additional conclusions can be drawn. Section 11 summarizes all our findings, drawing the last conclusions on the work, while, Section 10 discusses in depth the limitations of our current method and speculates on possible resolutions and relevant future work. Finally, an appendix with supplementary material referenced throughout the document has been included, following the Bibliography section.

Throughout the document the following terms will be used interchangeably: *deepfake* - *face forgery* - *modified* or *fake image* to represent any image that has been manipulated to hide the original identity. *Genuine image* - *real* or *authentic image* are terms used to represent samples of true identities. We will also be referring to *neural network* as simply *network* or *model*. In addition, the term *target* is used to indicate the sample whose face will be swapped with the identity of the *source* face (a different face image that will be used to modify the target image).

In summary, the contributions mentioned below are made in this work. We demonstrate the comparable generalization performance of CLIP, a zero-shot language-vision model, to CNN-based models fine-tuned on the task of deepfake detection, without the need for specific optimizations or fine-tuning processes, but simply by querying the model with adequate descriptions. We introduce a fine-tuning paradigm for language-vision models similar to CLIP, suited for datasets with a limited number of classes, that addresses the problem of collisions during training caused by the absence of unique

image-text pairs. The approach improves both in-distribution performance and generalization performance on other data distributions. The achieved results are State-Of-The-Art and comparable to the best available methods in the field. Finally, we suggest that sufficiently robust text features have the potential to significantly influence the overall performance of the fine-tuned vision model, leading to results comparable to the State-Of-The-Art, without relying on the pre-trained weights for the vision encoder.

All the code used for this research is available for research purposes on the following link.

2.1 Research question

Synthetic media presents significant ethical and security concerns. Most current detection approaches are unable to generalize properly to unseen or novel synthesis techniques. As they lack semantic understanding, they rely upon features learned from the visual artifacts of the training samples. Therefore the following question arises:

Can the generalization of face forgery detection be improved by exploiting the semantic understanding and generalization capabilities of pre-trained language-vision models?

With the above inquiry representing the main research question of this work, the following sub-questions arise, in an attempt to understand in more depth the capabilities of the proposed approach:

- Do language-vision models (CLIP in our case) innately possess knowledge of deepfakes and to what degree can they recognize them?
- Can the performance of the network be improved compared to the baseline (zero-shot) capabilities and how?
- How does our approach compare to the currently available SOTA face forgery detection methods?
- What limitations arise with this specific approach?

The first sub-question is addressed through various experiments conducted in Section 8.1 and partially in Section 8.2 where the default capabilities of the CLIP network are explored. The second sub-question primarily concerns the fine-tuning method proposed in Section 6, while the third sub-question is addressed in Section 8.3. Finally, Section 10 is dedicated to examining all the limitations encountered in order to answer the last sub-question, while also introducing possible solutions and relevant future work. All sub-questions are also addressed in Section 11, summing up our work.

3 Related work & Motivation

Since face forgery can have such a profound social impact, the forensics community has been very active in the continuous development of solutions. [Rossler et al., 2019] introduced a benchmark dataset of deepfake videos, which quickly became a widely utilized resource across literature, enabling a more precise evaluation of performance and comparison among the different methods.

In an effort to accelerate the development of novel face forgery detection approaches, [Dolhansky et al., 2020] released another challenging dataset for detecting manipulated media, known as the *DFDC* dataset. The dataset was used in a competition sponsored by Facebook, Microsoft, and other partners, offering prizes up to 500,000$ for the best-performing algorithm. Around the same time, [Li et al., 2020d] proposed the *CelebDF* dataset, which also became a very valuable resource for the forensics community.

In most algorithms, identifying the discrepancies and inconsistencies of faces, also known as *artifacts*, is the key to classifying images as real or fake. In work such as [Li et al., 2018, Yang et al., 2019, Haliassos et al., 2021] this task is modeled through the detection of biological artifacts, such as inconsistencies in eye blinking, head poses and mouth movement, along the temporal dimension, by concatenating sequential video frames. In contrast, [Tariq et al., 2019] suggest detecting fake GAN-generated images, as well as human-generated deepfakes, with different pre-processing techniques that analyze the statistical features of single images. In the work of [Zhang et al., 2019, Durall et al., 2020, Qian et al., 2020, Wang et al., 2020] the fact that CNN-generated images create anomalies in the frequency spectrum is exploited.

Although these approaches achieve impressive results when the inputs are of the same distribution as those used during training, they struggle to generalize to novel or different deepfake synthesis methods. For this specific reason, many authors shifted their attention to developing methods with generalization in mind, proposing techniques aiming to improve classifier detection on novel or different deepfake types than those trained on. [Khodabakhsh

et al., 2020] model the problem from the perspective of anomaly detection. The anomaly detector network is trained on pristine data only and learns to predict the conditional probabilities of observing a pixel given all pixels preceding it. The extracted features of the model are then used to train a simple classification model with good generalization capabilities across different deepfake synthesis methods. Similarly, [Tariq et al., 2020, Tariq et al., 2021] describe different data-driven methods that do not rely upon artifacts of specific generation techniques. In the first approach, a Convolutional LSTM-based Residual Network takes as input a sequence of consecutive video frames to learn temporal information, allowing it to detect unnatural-looking artifacts between frames. The second proposed approach expands upon the previous one by adding spatial information as well as temporal, yielding good results at the time, especially for high-quality deepfakes.

Another common approach across the literature for improving generalization is using data augmentation techniques. [Yang and Lim, 2020] propose a method to generate samples of a similar distribution to that of a given one-shot image example. The newly generated samples can later be used as augmented training samples for fine-tuning on the downstream classification task. [Chen et al., 2022a] propose to improve the generalization of deepfake detectors with the help of adversarial data augmentation and training, achieving great results for both low and high image resolutions.

In another line of work, the attention mechanism of Transformer networks is exploited for the generalization task. [Zhao et al., 2021] argue that the discrepancies between real and fake faces are usually very subtle, proposing a novel deepfake detection model architecture based on the attention mechanism. [Wang and Deng, 2021] utilize an attention-based data augmentation mechanism to guide the detector to refine and enlarge its attention. By "erasing" the most influential areas in an image, with respect to the attention matrix, the model is trained to capture other subtle inconsistencies which may have not been detected previously and may lead to correct classification.

Other solutions for the generalization problem include [Kim et al., 2021], where the authors propose a domain adaptation framework, with a teacher-student paradigm, used to mitigate the problem of catastrophic forgetting,

during fine-tuning on novel deepfake instances. This method improves computational efficiency while also preserving the original's model performance. In another interesting approach proposed by [Zhu et al., 2021], face images are disentangled into their 3D counterparts of geometry and lighting features, achieving great results on specific datasets and synthesis methods.

Some more recent publications expand and improve on the methods presented until now. For example, [Chen et al., 2022b] introduce yet another data-augmentation learning paradigm, based on synthesizing pseudo-training samples similar to the input image, which are used at test-time to fine-tune the model, before determining the final prediction. [Cao et al., 2022] propose another anomaly detection reconstruction approach, based on a joint reconstruction-classification training paradigm, under the assumption that the reconstruction of genuine faces can enhance the learned representations to be aware of forgery patterns, improving both the IN-distribution[1] and Out-Of-Distribution[2] detection tasks compared to the baseline model. [Zhuang et al., 2022] again exploit the attention mechanism of Transformer architectures to model an unsupervised training paradigm centered around detecting inconsistencies for single images. [Guan et al., 2022] adopt a local-to-global learning paradigm exploiting attention and temporal information within local image patch sequences. This approach is regarded as one of the best-performing methods for multi-frame deepfake detection in the current literature. However, the main limitation of this approach, is the performance uncertainty for single images, as experiments were carried out with sequential frames as input. Finally, [Shiohara et al., 2022] introduces a data augmentation-based training paradigm, that depends on synthetic images generated by blending together a pair of pristine images. To the best of our knowledge, it is the best-performing generalization approach currently developed for the purpose of single-image deepfake detection, not depending on a sequence of frames to capture temporal information. However, the training

[1] *In-Distribution* refers to samples with the same distribution as that of the training data

[2] *Out-Of-Distribution* (OOD) refers to samples with a different data distribution than what used for training. In our case, face forgeries types are not included in the FaceForensics++ dataset, Celeb-DF, and DFDC.

paradigm requires pairs of pristine images and their deepfake counterpart, together with landmarks of the face features to execute the blending between real images, which can be very limiting, making it very hard to apply to real-world data where such information is usually not available.

An alternative field of research for face forgery detection has focused on the contrastive learning paradigm. [Sun et al., 2022] introduce the *Dual Contrastive Learning* paradigm for the task of face forgery, based on constructing and comparing hard image pairs. [Dong et al., 2023] propose a mixed contrastive and data augmentation approach based on RGB and SRM features to try and improve textural and semantic information.

Despite the existence of a wide range of deepfake detection methodologies, to the best of our knowledge, none of them tries to leverage the semantic understanding of concepts around deepfakes through text and NLP. This raises an open question regarding whether the generalization abilities of pre-trained image-language models can be effectively utilized to enhance the detection of facial forgeries.

In the existing literature, there is a noticeable gap where deepfake detection methodologies have not explored the potential of leveraging semantic understanding of deepfake concepts through text and natural language processing (NLP). This raises an open question regarding the extent to which pre-trained image-language models can be effectively employed to improve the generalization capability and robustness of face forgery detection. Our work fills this gap in the existing literature by examining CLIP's potential for deepfake detection and proposing a novel fine-tuning method to improve its performance in this domain.

4 Background Work

4.1 Attention is all you need

Transformer models were introduced by [Vaswani et al., 2017] in 2017 for the task of machine translation and have since revolutionized the field of

natural language processing (NLP), quickly becoming the State-Of-The-Art architecture for many tasks, such as text generation and question answering.

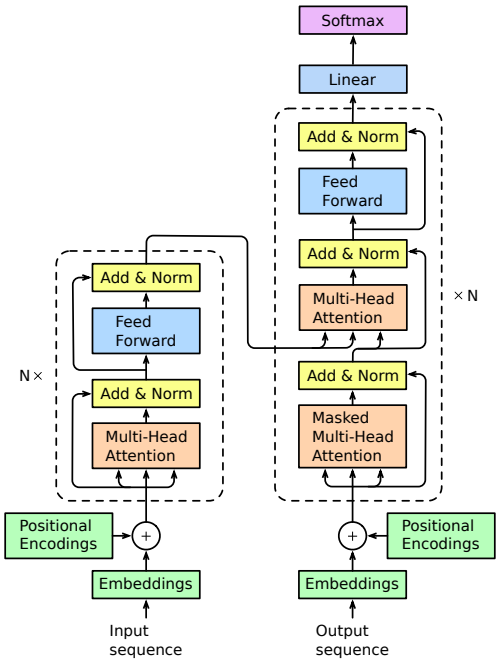


Figure 1: Architecture of the Transformer network. The left part of the figure represents the encoder stack, while the right side is the decoded part of the network.

4.1.1 Architecture

At the highest level, Transformer architectures are composed of an encoder and a decoder network and are based on the *self-attention* mechanism. Self-attention allows the network to weigh the importance of the different parts of an input sequence when making predictions. Unlike Recurrent Neural Networks (RNNs) and similar architectures created to process sequential data, Transformers are better at modeling long-range dependencies and have the ability to process the entire input sequence in parallel, making them extremely efficient during training. During inference, on the other hand, the model be-

haves in an auto-regressive manner and uses the previously generated values as additional input to generate the next one.

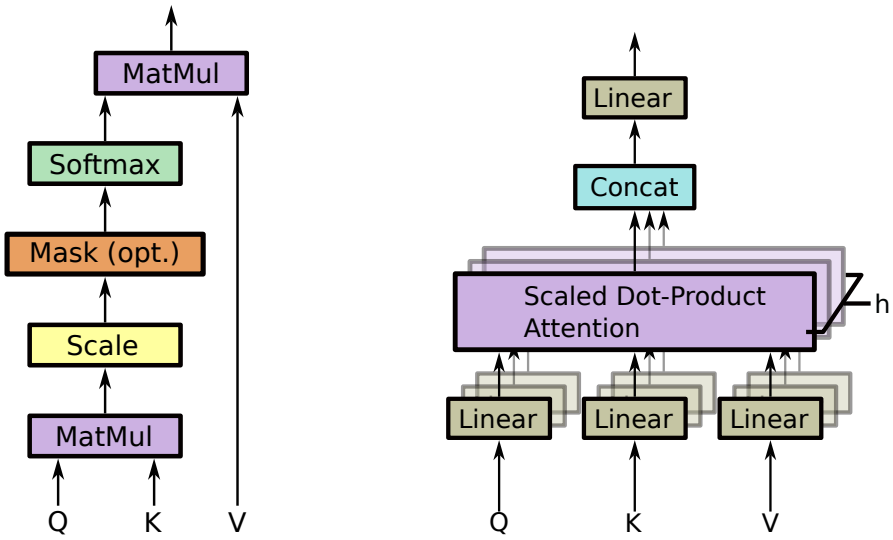
Given a sequence of inputs $x = (x_1, ..., x_n)$, the encoder learns to map them to a sequence of continuous representations $z = (z_1, ..., z_n)$. The encoded sequence is then fed to the decoder, which produces a sequence of tokens $y = (y_1, ..., y_n)$, one by one. A representation of the Transformer architecture has been included in Figure 1. Both the encoder and the decoder networks follow the overall architecture of using stacked self-attention layers followed by fully connected layers.

The **encoder** consists of N stacked layers. Every layer is composed of two sub-layers, the first being the *multi-head self-attention* layer and the second being a *position-wise fully connected feed forward* layer. Around every stack of sub-layers, a residual connection is added as described in [He et al., 2016], followed by a layer normalization as described in [Ba et al., 2016]. The overall output from each sub-layer can be therefore summarized by the following equation $LayerNorm(x + sublayer(x))$.

The **decoder** similarly to the encoder network is also a stack of N identical layers. Differently from the encoder, it introduces a third sub-layer that performs the multi-head attention on the output of the encoder and the currently generated sequence and is positioned between the first muti-head attention (attending to the decoded output) and the feed-forward layer. As in the encoder, a residual connection around each sub-layer is used, followed by a layer normalization. Differently from the encoder's attention, in the decoder we introduce the concept of *masked* attention, to prevent the mechanism to attend to positions following the one being predicted.

4.1.2 Attention mechanism

The attention function can be thought of as the process of understanding which part of the input the model should focus on. The attention mechanism used in [Vaswani et al., 2017], is called the *scaled dot-product attention*, and the process is represented in Figure 2a. The input consists in a set of queries q, keys k, and values v which are all embedded vectors of the input sequence.



(a) Scaled dot-product attention for a set of queries Q, keys K and values V.

(b) Multi-Head attention for h sets of linear projections of the queries Q, keys K and values V.

Figure 2: Representation of the attention mechanism used in Transformer networks.

The queries and keys have the same dimension d_k, while the values have dimension d_v. Since this operation is parallelizable, the attention function can be performed for multiple sets of (q, k, v) as one big operation between matrices (Q, K, V), resulting in the following equation:

$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}}) \cdot V \tag{1}$$

The term $\sqrt{d_k}$ is a scaling term used to normalize the dot-product when using larger latent representations d_k for the queries q and keys k.

In the original publication of [Vaswani et al., 2017], the authors found it effective to repeat the calculation of attention for a number h of projections of (Q, K, V). This process allows the Transformer to learn different inter-dependencies and relations between words. Given the dimensions of a single

attention function for queries, keys and values of dimension d_{model}, we can define the following matrices to represent the parameters used for the h projections of the input sequence: $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ where $i = 1, .., h$. Given the weight matrices the calculation of the complete multi-head attention is given by the two following equations:

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^W) \tag{2}$$

$$MultiHead(Q, K, V) = Concat(head_1, .., head_h) \cdot W^O \tag{3}$$

where $W^O \in \mathbb{R}^{h\dot{d}_k \times d_{model}}$ is the last linear layer as shown in Figure 2b, that takes the concatenated multi-head attention and projects it back to the correct dimension for the next layer.

4.1.3 Positional Encoding

The model requires positional information relative to the sequence, to make use of its order. That is achieved through two sine and cosine functions as below:

$$PE_{(pos,2i)} = sin(pos/10,000^{2i/d_{model}}) \tag{4}$$

$$PE_{(pos,2i+1)} = cos(pos/10,000^{2i/d_{model}}) \tag{5}$$

where pos represents the position, while i is the dimension.

4.1.4 Feed-Forward Network

The last sub-layer in the encoder and decoder blocks is a fully connected feed-forward layer, that consists of two linear layers with a RELU activation function in between. The input and the output dimension of the feed-forward layer is d_{model} and the hidden layer has a dimension d_{ff}. The output of the feed-forward network is represented by the following equation:

$$FFN(x) = max(0, x\dot{W}_1 + b_1) \cdot W_2 + b_2 \tag{6}$$

where the W_1 and W_2 represent the weight matrices of the two layers and b_1 and b_2 represent the bias terms related to the two layers.

4.1.5 Operation sequence

In this section, the high-level flow of data inside the Transformer network for a sequence-to-sequence task, such as translation from English to French, is described. Given a sequential input, each element of the sequence is embedded with the added positional encodings and is passed to the first encoder block. The encoder calculates the multi-head attention and adds the result to the output of the residual connection. The vector is normalized through layer normalization and is therefore passed through a feed-forward network. The output is yet again summed to the residual connection and layer-normalized. This routine is repeated N times, where N is the number of encoder blocks.

Once the input has been passed through all the encoder layers, a final connection connects the output of the last encoder block to every decoder block. The decoder starts the autoregressive process by attending to the output-generated embeddings, which, at the first iteration corresponds to the [SOS] token.

This layer is called the *Masked Multi-Head Attention* because during training, the whole output sequence is known and therefore its prediction can be parallelized. Passing the whole output sequence to the first sub-block of the decoder allows us to predict all the positions simultaneously. Consequently, masking has to be used when calculating the attention, to prevent the decoder from attending to parts of the sequence after the position being predicted. The masking is done by setting the values of the $Q \cdot K$ matrix corresponding to the connections expressing the relation of words in future positions to $-\infty$, before the softmax step. This method of training is called *Teacher Forcing* and is used to train efficiently Transformers while preserving the autoregressive capability of the model.

Consequently, the same operations as in the encoder's first sub-block are performed, until the next block, which is called the Eencoder-Decoder Attention" block. Here, attention is calculated between the queries of the pre-

vious decoder sub-block and the keys and values of the final encoder block. The output is then again added to the residual connection and everything is normalized and fed to the feed-forward sub-block. This step concludes the first iteration of the decoder block, which is then repeated N times.

The output of the last decoder block is passed to a linear layer, representing the size of the target vocabulary (e.g. the number of French words in the translation task previously mentioned), and a Softmax operation is applied, transforming the outputs into probability distributions, where each value represents the probability of the token in the vocabulary to be the next in the output sequence.

4.2 Large language models (LLMs) in computer vision tasks

Transformer architectures introduced from [Vaswani et al., 2017] have heavily revolutionized the field of Natural Language Processing (NLP), while also making their way into the field of computer vision, as explained by the survey of [Jia et al., 2022]. The works of [Dosovitskiy et al., 2020, Liu et al., 2021] can be considered some very popular examples of attention architectures developed for computer vision.

Another significant factor for NLP development has been pre-training methods, that learn directly from massive amounts of raw text, such as [Dai and Le, 2015, Peters et al., 2018, Devlin et al., 2018]. These methods, such as autoregressive and masked language modeling, have shown remarkable capabilities by leveraging task-agnostic objectives on large-scale datasets.

Subsequently, the development of the sequence-to-sequence paradigm [McCann et al., 2018, Radford et al., 2019, Raffel et al., 2020] further enabled task-agnostic architectures to transfer knowledge to downstream tasks, without specialized customization and fine-tuning. Training paradigms such as supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF) [Howard and Ruder, 2018, Stiennon et al., 2020, Köpf et al., 2023] have also been under development for NLP, making the networks more robust and adaptable to human-machine interactions.

On the other hand, common generalization techniques in computer vision have mainly depended on self-supervised learning approaches, where, models learn visual representations by solving pretext tasks on unlabeled images (e.g. context prediction, colorization, jigsaw puzzles, etc.) as in [Henaff, 2020, Chen et al., 2020b, He et al., 2020, Grill et al., 2020], usually done on big crowd-labeled datasets such as ImageNet [Deng et al., 2009]. Another very common approach for learning visual features is contrastive learning, where the embeddings of different image classes are pushed farther away, while images of the same class are pushed closer to each other in the latent space. [He et al., 2020, Chen et al., 2020a, Khosla et al., 2020, Han et al., 2021] are some example works in the field of contrastive learning.

While the above vision methods have shown success, they lack a semantic understanding of concepts as they rely primarily on visual patterns. Inspired by the success of pre-training methods in NLP, researchers have also explored the potential of vision-and-language pre-training. Early work by Mori et al. [Mori et al., 1999] proposed a two-process method, the first dividing images into sub-images with keywords, while the second carrying out the vector quantization of the sub-images. This method showed that each sub-image can be correlated to a set of words selected from words assigned to the original image. Quattoni et al. [Quattoni et al., 2007] presented a method for learning representations from large quantities of unlabeled images with associated captions. [Srivastava et al., 2017] proposed a learning approach for multi-modal data using Deep Boltzmann Machines. Joulin et al. in [Joulin et al., 2016] further investigated deep representation learning, by training multi-modal networks to predict a bag of words extracted from the captions of the YFCC100M [Thomee et al., 2016] image-caption dataset, with a weakly supervised learning approach. Overall, these studies demonstrated the usefulness of pre-training text for learning image representations.

Nevertheless, it is important to note that pre-training models using natural language supervision for image representation learning has been challenging. Although, [Li et al., 2017] showed promising results by predicting phrase n-grams related to images, the zero-shot performance on common benchmark datasets was relatively low compared to alternative approaches. In a similar

fashion, [Mahajan et al., 2018] tried to push the limits of weakly supervised learning by training models to predict Instagram hashtags on billions of images, resulting in an effective pre-training technique. For both the approaches mentioned above, however, the number of classes has to be defined *a priori* and lack a mechanism for dynamic outputs, limiting their generalization capabilities.

In some more recent work of [Li et al., 2020a], both visual and text contents are fed into a multi-layer Transformer for cross-modal pre-training, including multiple tasks to learn context-aware representations. In a different approach, [Li et al., 2020c] suggest using object tags detected in images as anchor points to significantly ease the learning of text-image alignments. As most of the previously cited methods require big amounts of data and training time, an effort to improve the training efficiency of language-vision models was done by [Desai and Johnson, 2021], who explored a training paradigm based on images and dense captions pairs. Similarly, [Zhang et al., 2022] explain ConVIRT, an unsupervised strategy to learn medical visual representations, by exploiting naturally occurring paired descriptive text, requiring a fraction of the data used by previous approaches in the medical field. Finally, one of the most known and recent language-vision training paradigms is the *Contrastive Language Image Pretraining (CLIP)* introduced by [Radford et al., 2021]. This model is a simplified version of the ConVIRT approach (which was in the pre-print stage when CLIP was being developed). Both CLIP and ConVIRT are trained to maximize the agreement between the true image-text pairs by using bidirectional losses. Overall, the field of language-vision models is still evolving, with ongoing research exploring different open roads.

5 Datasets

With the increasing emergency of AI-manipulated videos, several publicly available datasets have been contributed by academia and industry, in order to promote the development of deepfake detection solutions. Many of these datasets are also described in the work of [Almars, 2021]. For our experiments, the datasets presented below are used.

5.1 FaceForensics++ (FF++)

This benchmark dataset was proposed by [Rossler et al., 2019] and consists of 1000 video sequences manipulated with the following face forgery approaches: *Deepfake* [Karras et al., 2019a], *Face2Face* [Thies et al., 2016], *FaceSwap* [Dale et al., 2011, Garrido et al., 2014] and *NeuralTextures* [Thies et al., 2019].

The videos in the dataset were compressed with a H.264 codec, on two different compression levels. High-quality compression is denoted with *HQ*, indicating a constant quantization rate of 23, and low-quality compression *LQ*, created with a quantization parameter equal to 40. The *HQ* videos compose around 35% of the total dataset, around 50% are *LQ*, and the remaining ones are videos that were not compressed, denoted as *Raw*.

For every manipulated video sequence, the pristine source and masks indicating the pixels that were modified are also included in the dataset. The dataset was collected from YouTube videos of mostly frontal faces without any occlusion, allowing the creation of automated and realistic face forgeries. Finally, the dataset is fairly balanced with respect to gender.

5.2 Celeb-DF

This is a video benchmark dataset, proposed by [Li et al., 2020d]. It contains 590 pristine videos and 5,639 DeepFake videos (more than 2 million frames) generated with an improved version of the *DeepFaceLab* framework, proposed in [Perov et al., 2020]. The enhanced version of the algorithm used to create the deepfakes, improved the original resolution of the synthesized faces to 256x256 pixels, the color mismatch between the source and target faces, the accuracy of face masks for applying the deepfake, and the temporal flickering across frames.

The videos are sourced from YouTube and feature 59 celebrities of different genders, ages, and ethnic groups. The original videos come in various aspect ratios and resolutions, but the deepfake videos are standardized to a resolution of 256x256 pixels. All the videos are converted to MPEG4.0

format.

5.3 Deepfake Detection Challenge Dataset (DFDC)

The Deepfake Detection Challenge Dataset (DFDC) is, to the best of our knowledge, the largest dataset publicly available of face swap videos. It was created as a joint effort by *Facebook* (currently *Meta*) and other industry leaders, in order to accelerate the development of methods that tackled face forgery, by sponsoring a public Kaggle competition.

It is available in two formats, the Preview and the Full dataset, developed from [Dolhansky et al., 2019, Dolhansky et al., 2020]. The Preview version consists of 5,000 videos and features two modification algorithms while the Full dataset contains 124,000 videos created with eight modification algorithms. The video sequences were created by 3,426 paid actors that gave their consent to have their faces manipulated by machine learning techniques. The source videos were pre-processed with face tracking and the faces resized to 256x256 pixels. The target videos were generated with multiple face swap methods, including FSGAN [Nirkin et al., 2019] and StyleGAN [Karras et al., 2019b].

6 Research methods

6.1 Original CLIP pretraining method

The general idea of the Contrastive Language Image Pre-training (CLIP) paradigm is to train a text encoder and an image encoder to produce similar embeddings.

Let N denote the size of a batch of images $x_i \in \mathbb{R}^{N \times C \times H \times W}$, aligned with the corresponding batch N of text descriptions $x_t \in \mathbb{R}^{N \times T}$. C represents the channel dimension, H the height and W the width of the image, while T is the sequence length of the tokenized text. *Aligned*, meaning that every image and text pair have the same index in the corresponding batches. The first step towards training our model is to obtain feature representations of

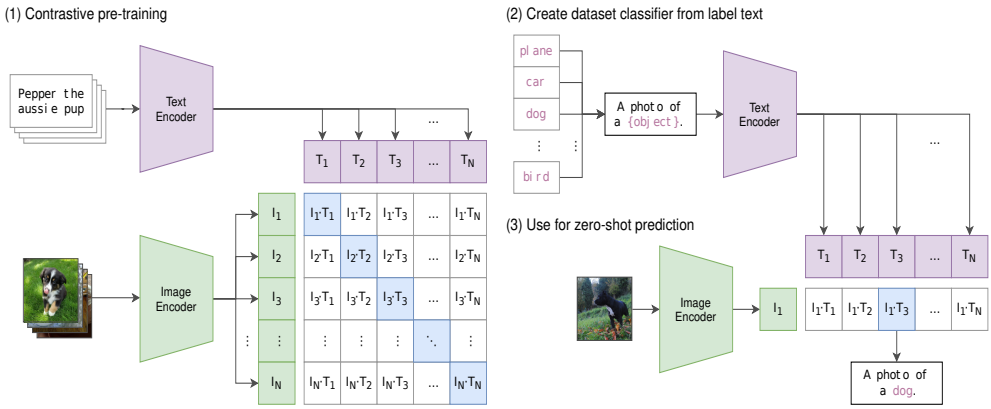


Figure 3: Figure taken from the original work of [Radford et al., 2021] where the CLIP model is introduced. Sub-figure (1) represents the contrastive pre-training where the model's image and text encoders are trained jointly to produce similar embeddings, by learning to predict the correct image-text pairs of a batch of images and descriptions.
Sub-figures (2) and (3) represent the inference step, where a zero-shot classifier is constructed by embedding all the descriptions and predicting which of them is more similar to the image embedding.

the data by passing it through two independent encoders characterized by the following notations:

$$E_i : \mathbb{R}^{N \times C \times H \times W} \rightarrow \mathbb{R}^{N \times c \times h \times w} \tag{7}$$

$$E_t : \mathbb{R}^{N \times T} \rightarrow \mathbb{R}^{N \times S} \tag{8}$$

Notation 7 refers to the image encoder, while notation 8 refers to the text encoder. S is the dimension of the extracted text features and c, h and w are the image feature dimensions. We represent the outputs of this step as $E_i(x_i)$ for the image features and as $E_t(x_t)$ for the text features.

Now let $X_i = c \times h \times w$ represent the dimension of the features of a single image, collapsed on one dimension. We can therefore use two linear layers, $W_i \in \mathbb{R}^{X_i \times M}$ and $W_t \in \mathbb{R}^{S \times M}$ to project our extracted features to a common dimension M.

The transformation to the common dimension is done by taking the L2 norm of the dot product between the encoded inputs and the weight matrices as such:

$$I_e = ||E_i(x_i) \cdot W_i|| \in \mathbb{R}^{N \times emb} \tag{9}$$

$$T_e = ||E_t(x_t) \cdot W_t|| \in \mathbb{R}^{N \times emb} \tag{10}$$

where I_e and T_e represent the image and text embeddings projected on the common dimension M. Finally, given the temperature parameter t optimized during training, the similarities can be calculated with the dot product between the embeddings, with the following formula:

$$Sim = (I_e \cdot T_e) \times e^t \in \mathbb{R}^{N \times N} \tag{11}$$

Let now $L = diag(N)$ be a diagonal matrix of shape $N \times N$ with the correct image-text pairs in its diagonal. In order to calculate the bidirectional loss we first calculate the image and text cross-entropy losses separately, as below:

$$L_{img} = -\frac{1}{N} \sum_{n=1}^{N} \left[\sum_{d=1}^{N} (l_{nd} \times \log s_{nd}) \right] \tag{12}$$

$$L_{txt} = -\frac{1}{N} \sum_{n=1}^{N} \left[\sum_{d=1}^{N} \left(l_{nd} \times \log s_{nd}^T \right) \right] \tag{13}$$

where, L_{img} represents the image part of the loss and L_{txt} the text loss. The equations are almost identical. The outer sum goes over all images in the batch and the second summation runs over all classes, or in our case, the text descriptions. The term l_{nd} represents the correct label for image n and description d, s_{nd} represents the similarity for image n and description d. The difference in the two equations lies in the term s_{nd}^T, which represents the elements of the transposed similarity matrix Sim^T. This is possible since our Sim matrix is square and the label matrix is diagonal, creating the bidirectional loss. To complete the process and allow for the optimization step, we calculate the mean of L_{img} and L_{txt}, as shown in Equation 14 below.

$$loss = (L_{img} + L_{txt})/2 \tag{14}$$

A visual representation of the original CLIP method can be found in Figure 3.

6.2 Finetuning CLIP for deepfake detection

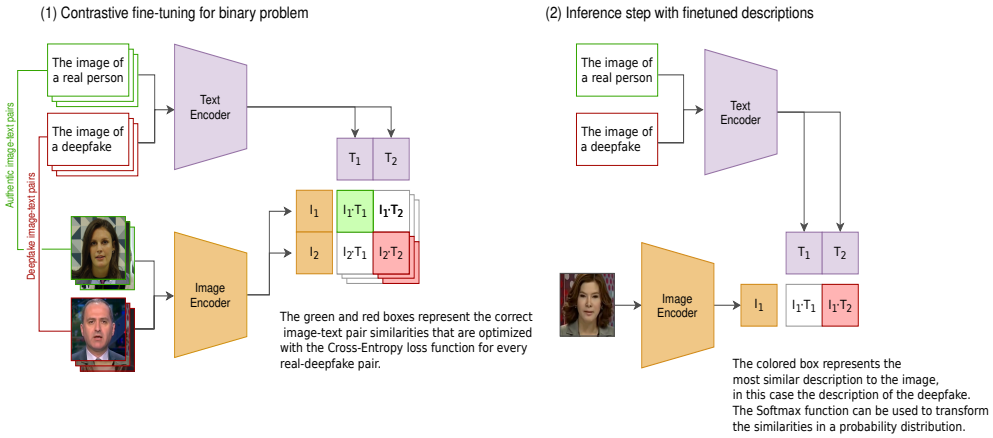


Figure 4: The Figure represents the adaptation of the CLIP paradigm when fine-tuned on a binary problem. Sub-figure (1) depicts the fine-tuning process, where real and deepfake pairs of image-text sets are used to optimize the similarity of the embeddings related to the two classes. Sub-figure (2) depicts the inference process, where the same labels as those optimized in the fine-tuning step (1) are used to collect predictions.

While some recent work exists on efforts to fine-tune zero-shot models on traditional classification downstream tasks, they mainly focus on using teacher-student training paradigms where CLIP is the teacher model or fine-tuning only the vision encoder.

For example, [Wang et al., 2022] propose a teacher-student method to distill knowledge from the pre-trained CLIP model (teacher) on existing architectures (student). Although this method does not directly fine-tune the language-vision model, it shows promising results both in few-shot[3] and fully-

[3] *few-shot learning* refers to training or fine-tuning a model on a low number of samples

supervised learning settings.

In the work of [Dong et al., 2022], the authors fine-tune the vision encoder of CLIP on the ImageNet dataset and argue that, although special attention is required when choosing the hyper-parameters and in the fine-tuning procedure, CLIP itself can be a strong fine-tuner and achieve SOTA results.

In another approach described in the work of [Wortsman et al., 2022], the authors argue that when the visual encoder of language-vision models is fine-tuned, generalization performance is lost under distribution shifts and propose a weight ensemble method, between the pre-trained weights and the fine-tuned weights, achieving higher robustness while preserving the performance in the in-distribution data.

However, since the text features for deepfake detection do not appear to be totally refined yet, as it will be shown from the experiments of Sub-sections 8.1 and 8.2, a method is required to push higher the overall performance of the model when using the correct descriptions associated with the task.

In some more recent work, the authors of [Goyal et al., 2023], experiment with *linear probing*[4] and full fine-tuning of image and text, similar to what required from our problem setting, discovering that by also fine-tuning the text encoder, it was possible to gain additional performance on the overall results. A limitation that this work mentions is that collisions of the same class are possible in the same mini-batch during training. However, even with collisions, their fine-tuning method was able to outperform the baseline. As an effort to further improve performance, they experiment with masking out the terms in the loss relative to collisions, however, resulting in decreased performance of the model.

In order to finetune the CLIP pre-trained model with the text encoder, we use a different approach from the one presented above, suited for our downstream task of deepfake detection, addressing the limitation of collisions in the batch. In addition, similarly to what is done in [Goyal et al., 2023], our method uses the contrastive loss utilized in the original CLIP pre-training method of

[4] *linear probing* refers to learning a linear classifier on top of the features of another model.

[Radford et al., 2021] and can sample different descriptions related to the same class.

To make our method work, the model is fed with a batch n of pairs of images (with their descriptions), one of a genuine and one of a manipulated face, and the contrastive loss is calculated separately for each of the pairs, as done in the original CLIP pre-training method, but with only two samples per batch. Assuming a batch size N of 2, for Equations 12 and 13, we can use them to calculate n loss terms, one for each pair and apply Equation 14 to calculate the total loss for each pair. Finally, the mean over all the losses of the batch is calculated for the optimization step, resulting in the following loss:

$$loss = \frac{1}{n} \sum_{i=1}^{n} \left[(L_{i_{img}} + L_{i_{txt}})/2 \right] \tag{15}$$

A representation of the modified method used for deepfake detection is available in Figure 4.

This approach can easily be extended to accommodate any limited number of classes N, through two methods. The first one follows a similar process as the one described above, with the distinction that, at each step, two image-text pairs of different classes are sampled randomly from all the possible classes. Optimization is then possible in the same way as what is presented in the above binary adaptation of CLIP. A slightly different approach, potentially more computationally efficient, would be to use a subset n of the original classes N, as to calculate and optimize similarities between n unique classes instead of only two, at every step. The approaches mentioned are very similar to the original pre-training method but are adapted to work with a limited and arbitrary number of classes instead of unique image-text pairs, avoiding collisions.

As far as the descriptions are concerned, this approach can work with completely different descriptions for each image, a batch of descriptions for each class, or simply one description per class, allowing a lot of flexibility. For the second case, where we have a batch of descriptions for every class, we can sample a different description at each training step.

7 Experiemntal Setup

In order to assess the generalizability capabilities of our models, the FaceForensics++ (HQ) dataset was used for training, and generalization performance was assessed on Celeb-DF and DFDC unless otherwise specified. This approach aligns with previous practices in the field, allowing for direct comparisons with other State-Of-The-Art methodologies in the subsequent experiments.

To properly utilize the datasets for both training and evaluation, several preprocessing steps were undertaken, to convert the data from short videos to individual images, as required by our method. For the FaceForensics++ (HQ) dataset, 15% of the videos were allocated for testing purposes. In contrast, the train and test video splits for the *Celeb-DF* and *DFDC* (Preview) datasets were pre-defined and provided with the structure of the datasets.

To ensure consistency across all datasets, a frame-by-frame filtering process was applied to extract face images from every video sample. The procedure involved employing the *Dlib* Python library [King, 2009] through a two-step operation. Initially, the *Dlib* face detection algorithm was utilized to detect faces within each frame. Only frames where the face was predicted with an accuracy exceeding 95% were retained. Subsequently, the bounding boxes generated by the face detection algorithm were utilized to filter out any face with a resolution lower than 95x95 pixels. The resulting count of samples from these operations is outlined in Table 1 for the FaceForensics++ dataset and in Table 2 for the Celeb-DF and DFDC datasets.

Due to the substantial volume of collected samples, a constraint was applied for training our configurations, limiting the number of samples to 10,000 genuine images and 10,000 deepfake images. These images were randomly and uniformly sampled from the respective subsets of the dataset for both training and evaluation purposes. Specifically, a total of 10,000 images from the *youtube* subset and 2,500 images from each deepfake subset mentioned in Table 1 of the FaceForensics++ dataset were selected. For model testing following the training phase, 40,000 images were sampled from every dataset, equally split into 20,000 real and 20,000 deepfake images. These images were

FaceForensics++				
	Real		Fake	
Dataset	Train	Test	Train	Test
youtube	139,908	23,379		
Deepfake			138,200	23,130
Faceswap			111,361	18,421
Face2Face			138,574	22,864
NeuralTexture			110,955	18,719

Table 1: Dataset splits for the FaceForensics++ dataset. The values reported are the result of a train and test video split of 85% and 15% respectively, following a filtering operation, with the *Dlib* python library. The filtering operation consisted of two steps. Firstly, we filter out the frames that were predicted to contain a face with an accuracy lower than 95% from the *Dlib* network, and secondly, by using *Dlib*'s predicted bounding boxes around face images, we filter out any frames with faces smaller than 95x95 pixels.

utilized to calculate the statistics presented in the experiment tables.

The CLIP model utilized in this study is the *ViT-L/14* architecture, as originally presented by OpenAI in their publication [Radford et al., 2021]. This model comprises approximately 470 million parameters. The image encoder adheres to the architectural design of the *ViT-Large* model proposed in [Dosovitskiy et al., 2020], employing 14x14 patches and incorporating an additional layer normalization step after combining the positional encodings with the embeddings. In this work, the vision encoder consists of N layers, specifically 12 in our case, with a dimension d_{model} of 1024. The MLP hidden size d_{ff} is set to 4096, and the model employs 16 attention heads h. Before being passed into the Vision Transformer, the images undergo preprocessing transformations. This involves resizing the images to 224 pixels with bicubic interpolation, followed by a center crop of dimensions 224x224. Subsequently, the images are rescaled to values between 0 and 1 and nor-

	Celeb-DF		DFDC	
	Train	Test	Train	Test
Real	289,716	70,154	561,044	68,971
Fake	2,013,957	130,659	2,874,521	130,659

Table 2: Dataset splits for the Out-Of-Distribution datasets. Celeb-DF and DFDC already provide the train and test splits of the videos. All the videos were filtered similarly to what was done for the FaceForensics++ dataset.

malized using the mean values (0.481, 0.457, 0.408) and standard deviation values (0.268, 0.261, 0.275).

The text encoder in our network adheres to the architecture outlined in [Vaswani et al., 2017], incorporating the modifications proposed in [Radford et al., 2019]. Notably, the layer normalization is relocated to the beginning of each sub-layer, and an additional layer normalization is introduced after the final self-attention layer. Moreover, the weight initialization is scaled by a factor of $1/n$, where n represents the number of layers with residual connections. The Transformer employed consists of N layers, specifically 12 layers in our configuration. It operates with a dimension d_{model} of 768, an MLP dimension d_{ff} of 3072, and utilizes 8 attention heads h. Each text description undergoes encoding using a *lower-cased byte pair encoding* (BPE) approach, employing a vocabulary size of 49,152. Additionally, a maximum sequence length is imposed, of 76 tokens. The resulting sequences are enclosed with [SOS] (start of sequence) and [EOF] (end of sequence) tokens.

Regarding the text descriptions utilized for training our configurations, they were not included in any of the datasets and were therefore generated manually. For configurations denoted with the *"1desc"* suffix, the following description was employed for genuine face images: "The image of a person." For deepfake images, the description used was "The image of a deepfake." In contrast, configurations labeled with the *"10desc"* and *"550desc"* suffixes correspond to the *multi-description* setups. These configurations involve a total of 10 and 550 pairs of real and fake descriptions respectively, which

were sampled using ChatGPT and subsequently refined manually where necessary. The main objective of these descriptions was to encompass a general description of genuine and deepfake images, ensuring sufficient coverage of semantic features. The specific content of the descriptions was not critical, as long as they captured the essence behind the concepts of genuine and deepfake images. The prompts employed to generate the 1100 descriptions can be found in Appendix A, specifically in Table 9. Additionally, all descriptions are available in the GitHub repository associated with this document. Moreover, the descriptions for the *10desc* configuration were manually selected from the descriptions within the *550desc* configuration. The details of the chosen descriptions for genuine and deepfake images of configuration *10desc* can be found in Table 10 and Table 11 in Appendix A.

During the training of multi-description configurations, a sampling strategy is employed where two descriptions (one for genuine and one for manipulated images) are randomly selected from our lists for each batch of images. These descriptions are then appended to the corresponding real-deepfake image pairs. Consequently, the optimization step is performed similarly to having one description per class, but with multiple descriptions contributing to the training of the model. For testing and inference, the procedure for the *1desc* configurations is straightforward. The same descriptions used during training are employed, and the probability distribution over the logits for each image is calculated using the Softmax function. However, for the multi-description configurations, a different evaluation approach is implemented. Firstly, the logits of each image are computed for all the descriptions used during training. Subsequently, the mean value of the genuine and deepfake logits is calculated separately. Following this step, the probability distributions for the image being real or fake can be determined, similar to the *1desc* configurations, using the Softmax function. Furthermore, the multi-description configurations were evaluated using top-k sampling. After gathering the logits from all descriptions, the top K highest values are retained, and only those logits are considered for the final class prediction.

Due to computational limitations, the tuning of hyperparameters was conducted with limited prior experimentation. All configurations were trained

using a batch size of 16 and the Adam optimizer with a high decoupled weight decay value of 0.2, following the approach described in [Loshchilov and Hutter, 2017]. The beta values for the optimizer were set to (0.9, 0.98), and the epsilon value was fixed at 1e-6. During training, the learning rate was dynamically adjusted using a linear warm-up and cosine annealing scheduler. Specifically, a warm-up phase consisting of 10,000 steps was implemented to linearly increase the learning rate from 0 to 6e-7 for the *1desc* configuration and up to 8e-7 for the multi-description configurations. These learning rate values were determined to yield the best performance. Subsequently, the cosine annealing was carried out over 10,000 steps. In each warm-up and cosine annealing cycle, the peak learning rate was set to half of the previous cycle's peak learning rate. The training process included a stopping criterion based on the validation loss. If the validation loss did not improve by a minimum of 5% over three consecutive epochs, training was halted. This threshold was determined empirically during the hyperparameter tuning phase. For the final tests, the weights selected were those associated with the lowest validation loss value on the FaceForensics++ dataset during training.

In Section 8.1, an Xception network was trained [Chollet, 2017] as an additional baseline comparison, as it is widely utilized in face forgery detection. Our specific implementation was pre-trained on the ImageNet dataset, achieving a top-1 precision of 78.89% and a top-5 precision of 94.29%. The training process of the Xception model closely resembled that of the *1desc* configuration, with the adjustment in the weight decay of the optimizer, lowered to 1e-5.

All experiments were conducted on an RTX 2080 Ti GPU, with each configuration requiring approximately 15 hours to train using our current implementation. To ensure the reliability and consistency of the results, every experiment was repeated three times using different random seeds. The reported results represent the best performance obtained among the multiple runs.

The main evaluation metric used in our experiments is the Area Under the Curve (AUC) score of the Receiver Operating Characteristic (ROC) curve. The AUC score is a comprehensive metric for evaluating the generalization

capabilities of a model in a binary classification setting, as it provides a measure of the model's ability to distinguish and separate two classes. To comprehend the AUC metric, it is necessary to understand the ROC curve. The ROC curve illustrates the performance of a binary classifier across various discrimination thresholds. It is created by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at different threshold settings. The AUC value represents the area under the ROC curve, which serves as an indicator of the classifier's discriminatory power. A perfect classifier achieves an AUC value of 1, indicating a perfect separation between the two classes, while a value of 0.5 suggests that performance is equivalent to a random classifier. To enhance readability in the upcoming tables, the AUC values are presented as percentages (*AUC%*), obtained by multiplying the AUC score by 100. This conversion facilitates the interpretation of results.

Furthermore, in certain experiments, Precision is also reported as a supplementary metric. Precision is calculated by dividing the number of true positives by the sum of true positives and false positives.

8 Experiments & Results

8.1 Zero-shot experiments

Due to the lack of public disclosure regarding the data on which CLIP was trained, the model was treated as a black box to explore its potential in the deepfake detection task.

To assess CLIP's semantic understanding of deepfakes, some initial inference experiments were conducted using the default weights provided by OpenAI, on the FaceForensics++, Celeb-DF, and DFDC datasets. The results of these zero-shot experiments are summarized in Table 3.

The zero-shot experiments of Table 3 indicate that CLIP performs similarly to the Xception architecture fine-tuned from ImageNet weights, on the CelebDF and DFDC datasets. However, there is a noticeable performance difference on the FaceForensics++ dataset, which is expected since the Xception model was specifically trained on that data distribution. Furthermore, a comparison

between the "1desc" and "550desc" configurations reveals a 5% performance improvement for the "550desc" configuration, achieving an average AUC% of 64.16% compared to 59.23% for the "1desc" configuration.

Figure 5 on the other hand, provides an overview of the CLIP model's behavior when top-k sampling is applied to the 1,100 total labels in the "550desc" configuration. Performance tends to improve as more descriptions are considered. The performance pattern remains consistent across datasets, with the best results observed on the FF++ dataset, followed by the DFDC dataset, and Celeb-DF consistently posing the most significant challenge for the CLIP pre-trained network, when queried with multiple descriptions. Additional details and calculated values regarding the top-k experiment can be found in Table 12 in Appendix B.

Method	AUC %			AVG%
	FF++	CelebDF	DFDC	
Xception	**98.41**	53.43	60.78	70.87
CLIP (zero-shot - 1desc)	63.99	**58.43**	55.27	59.23
CLIP (zero-shot - 550desc)	78.11	52.13	**62.29**	64.18

Table 3: Benchmark experiments on the zero-shot capabilities of CLIP on the deepfake detection task.

As observed in Table 3 and Figure 5, the performance of CLIP depends heavily on the descriptions used. In the *1desc* configuration CLIP achieves a higher AUC% value than the DFDC dataset, which is not the case for the *550desc* configuration. This behavior is also observed in the qualitative experiment represented by Figure 6 and Table 4. The experiment consisted in sampling a random genuine face and a deepfake and observing the performance under two different sets of real-fake descriptions. The text descriptions and their relative performance are described in Table 4 while the images used can be found in Figure 6. The experiment showcases the fact that although the semantic meaning of the two pair of descriptions is similar, the results obtained when querying the CLIP model are completely opposite.

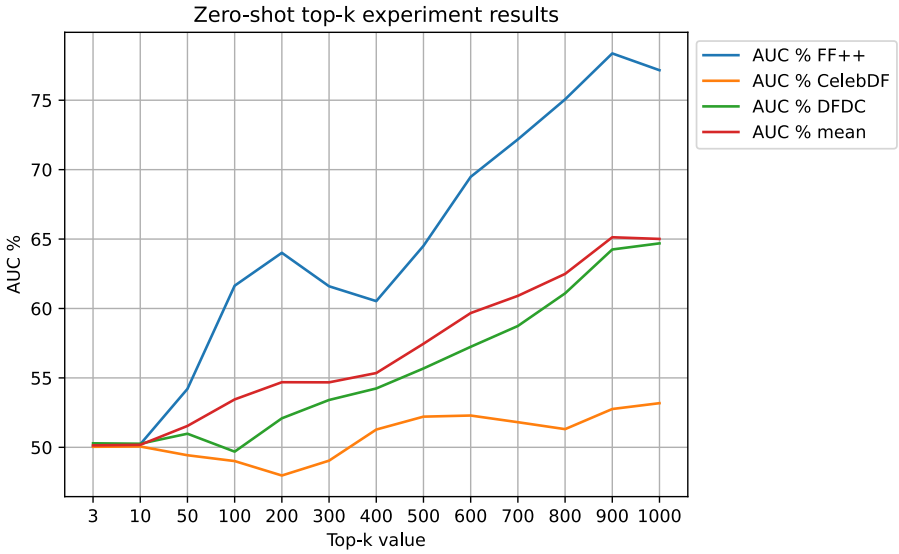


Figure 5: Zero-shot performance of the pretrained CLIP model with the *550desc* descriptions configuration. The plot shows that the performance improves by increasing the number of descriptions considered.

8.2 Vision encoder reset experiments

To better perceive the contribution of the text encoder's semantic understanding, several fine-tuning experiments were conducted on CLIP, with the vision encoder weights re-initialized. The weights of the text encoder were kept frozen during training to preserve the initial semantic understanding and structure. The aim of this experiment was to maximize the influence of the text descriptions on the network while minimizing the contribution of the pre-trained visual encoder. The configurations were named after the description pairs used during training and are as follows:

- **1desc**: The descriptions used are "The image of a person." and "The image of a deepfake.", as described in Section 7.
- **550desc**: This experiment used the 550 descriptions explained previ-

(a) Image of a genuine non-manipulated face, sampled from the FaceForensics++ dataset.

(b) Image of a face manipulated with the Deepfake technique, sampled from the FaceForensics++ dataset.

Figure 6: Samples used to qualitatively test the effectiveness of descriptions for the task of deepfake detection.

Method		Real Image		Fake Image	
		% Real	% Fake	% Real	% Fake
Zero-shot	bad descriptions	7.37	**92.63**	**62.60**	37.38
	good descriptions	**87.30**	12.68	8.15	**91.85**

Table 4: Performance of the zero-shot CLIP model with different descriptions. The *bad descriptions* represent the descriptions that lead to the wrong classification of the samples and are the following: *"Facial details in the image appear consistent and realistic, suggesting an authentic portrayal."* and *"The image shows signs of facial manipulation, giving the impression of an artificially altered appearance."*. The good descriptions, on the other hand, lead to a correct classification of the samples and are the following: *"The image showcases genuine facial features, reflecting an authentic representation."* and *"The image has been generated artificially, raising doubts about its authenticity."*.

ously, as it showed the best performance in the zero-shot experiments.

- **catDog**: The descriptions used were "The image of a cat." for real face images and "The image of a dog." for deepfake images.
- **carMoto**: The descriptions used were "The image of a car." for real face images and "The image of a motorcycle." for deepfake images.
- **FlakeBall**: The descriptions for this configuration were "The image of a paintball." for real face images and "The image of a cornflake." for deepfake images.

The results summarized in Table 5 show that the text descriptions used in training can have a significant impact on the features learned by the visual model, thus influencing the overall performance. The best-performing configuration is the *carMoto*, with more than 5% increased AUC% over the *1desc* and *550desc* configurations reaching the second and third best performance respectively. Differently from the zero-shot experiments, the *1desc* configuration performs slightly better than the multi-description configuration.

Method	AUC %			AVG %
	FF++	CelebDF	DFDC	
Xception	**98.41** [1]	53.43	60.78	70.87
CLIP (zero-shot - 1desc)	63.99	58.43	55.27	59.23
CLIP (zero-shot - 550desc)	78.11	52.13	62.29	64.18
CLIP (FlakeBall)	86.59	64.65	61.76	71.00
CLIP (catDog)	79.48	58.50	59.39	65.79
CLIP (carMoto)	**97.79** [2]	**87.41** [1]	**78.09** [1]	**87.76** [1]
CLIP (1desc)	**97.39** [3]	**74.79** [3]	**72.10** [2]	**81.43** [2]
CLIP (550desc)	95.48	**77.41** [2]	**69.95** [3]	**80.95** [3]

Table 5: Experiments on the CLIP model with reset vision weights and frozen text encoder weights. The first two rows are the experiments seen in Table 3 and have been reported here as a comparison. The last column reports the average performance between all three datasets. The descriptions for each fine-tuned CLIP configuration can be found in Sub-section 8.2.

8.3 State-Of-The-Art comparison

For the experiments in this section, both the image and text encoders were fine-tuned to compare the proposed approach with other State-Of-The-Art (SOTA) methods. Table 6 presents the results of the cross-dataset evaluation.

The results show that the *1desc* configuration, which uses a single description, outperforms the multi-description configurations, aligning with the findings in Section 8.2. The *1desc* configuration also exhibited faster convergence during training compared to the multi-description configurations, making it more computationally efficient. Additionally, the "carMoto" configuration trained in Section 8.2 achieves similar performance to the fully fine-tuned *1desc* configuration, indicating that it is possible to achieve high performance without tuning the text encoder or relying on pre-trained visual features.

The results on the out-of-distribution datasets show that CLIP can achieve SOTA performance, comparable to some of the best methods available, such as [Shiohara et al., 2022, Guan et al., 2022, Li et al., 2020b]. However, on the in-distribution dataset, CLIP's performance remains slightly below the average. The detailed performance on the in-distribution dataset can be found in Table 13 of Appendix B. The performance is fairly balanced with respect to the manipulation techniques, with the *NeuralTexture* one being the most challenging. However, it is important to note that the focus of this work is primarily on performance across different data distributions, rather than in the specific In-Distribution task.

The FaceForensics++ dataset was further utilized to conduct experiments using the Leave-Out-One method. In this experiment, the model was trained on all manipulation methods of the FF++ dataset, except one, and evaluated on the excluded method. The summarized results are presented in Table 7, where each AUC% column represents the deepfake detection dataset used for cross-evaluation, excluded from the training phase.

The experiment yields highly positive results, with the proposed CLIP method outperforming most other state-of-the-art (SOTA) approaches. Averaging over the datasets, the SBI method achieves a mean AUC value of 99.64%,

Method	AUC %			AVG%
	FF++	Celeb-DF	DFDC	
Face X-ray [Li et al., 2020b]	99.17	80.58	**80.92** [3]	86.89
OST [Chen et al., 2022b]	98.2	74.8	**83.3** [2]	85.43
DCL [Sun et al., 2022]	99.30	82.30	76.71	86.10
UIA-ViT [Zhuang et al., 2022]	99.33	82.41	75.80	85.84
SBI [Shiohara et al., 2022]	99.64	**93.18** [1]	**86.15** [1]	**92.99** [1]
LTTD * [Guan et al., 2022]	99.4	**89.3** [2]	80.4	**89.7** [2]
CRGB [Dong et al., 2023]	99.3	82.3	73.3	84.96
CLIP (1desc)	98.68	**86.68** [3]	79.09	**88.15** [3]
CLIP (10desc)	97.81	80.26	76.45	84.84
CLIP (550desc)	97.71	80.84	76.03	84.86
CLIP (carMoto)	97.79	87.41	78.09	87.76

Table 6: Cross-dataset evaluation experiments of the fully fine-tuned CLIP models compared with other SOTA methods. The results of each specific approach have been taken directly from their original publication. The last row corresponds to the vision reset experiment presented in Section 8.2 and has been reported here for comparison, as it provided one of the best results. The highlighted results indicate the top three configurations for each dataset. Methods with a * next to their names indicate multi-frame methods that evaluate each video as a whole.

while CLIP closely follows with an average of 96.27%. The next best-performing method, UIA-ViT, attains an average AUC value of 86.1%.

It is worth noting that the most challenging dataset for CLIP was the one generated using the NeuralTexture manipulation method. This aligns with the findings on the In-Distribution dataset presented in Table 13 of Appendix B, which demonstrates the challenging nature of the NeuralTexture manipula-

tion method. Overall, the results obtained from this experiment are consistent with the performance observed in the previous experiment of Table 6, with the CLIP method closely matching the performance of the best-performing SOTA approaches.

Method	AUC %			
	DF	F2F	FS	NT
LTW [Sun et al., 2021]	92.70	80.20	64.00	77.30
DCL [Sun et al., 2022]	94.90	82.93	-	-
UIA-ViT [Zhuang et al., 2022]	**96.70** [3]	**94.20** [3]	**70.70** [3]	**82.80** [3]
SBI [Shiohara et al., 2022]	**99.99** [1]	**99.88** [1]	**99.91** [1]	**98.79** [1]
CRGB [Dong et al., 2023]	94.10	81.40	65.60	79.20
CLIP (1desc)	**98.58** [2]	**98.49** [2]	**97.71** [2]	**90.33** [2]

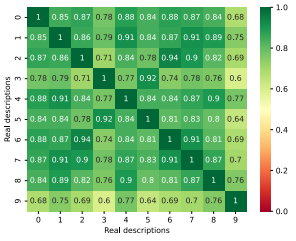
Table 7: Leave-Out-One experiment, where each column represents the dataset left out of training and evaluated on: DF: Deepfake, F2F: Face2Face, FS: FaceSwap, NT: NeuralTexture. The results of other methods were taken from their original publications.

8.4 Similarity of embeddings

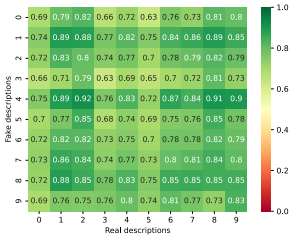
In this section, we conduct an analysis of the *10desc* configuration, which was trained in Section 8.3, in order to investigate the behavior of the text embeddings, relative to the descriptions used during the fine-tuning process. The descriptions relevant to the configuration have been included in Table 10 and Table 11 of Appendix A. The primary objective of this experiment was to examine the behavior of the embeddings by measuring their cosine similarity before and after the fine-tuning procedure. The results were calculated and visualized in the form of heatmaps in Figure 7. The mean values of the plots have also been summarized in Table 8 to facilitate the comparison between heatmaps.

The first row of heatmaps in Figure 7 represents the cosine similarities of the embeddings prior to the fine-tuning process. From the visualizations, it is evident that both the real and fake descriptions exhibit considerable similarity with each other.

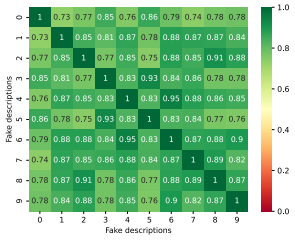
Furthermore, by examining the embeddings after the fine-tuning process in the second row of the heatmaps, we observe that the patterns detected before fine-tuning mainly remain preserved, with some subtle differences. Specifically, we notice a slight decrease in the intensity of similarities between the embeddings of real and fake descriptions. Additionally, we observe that the embeddings corresponding to descriptions of real faces exhibit a higher degree of similarity compared to the non-fine-tuned similarities. Similarly, the similarity of the embeddings related to the deepfake descriptions manifests an increase after fine-tuning.



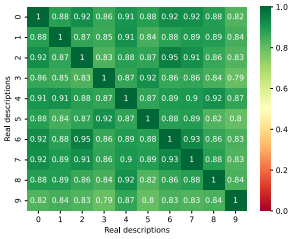
(a) Cosine similarity of the real descriptions for the zero-shot CLIP model.



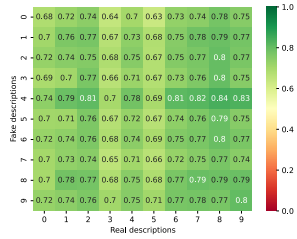
(b) Cosine similarity of the zero-shot CLIP network, between genuine descriptions on the x axis and deepfake ones on the y axis.



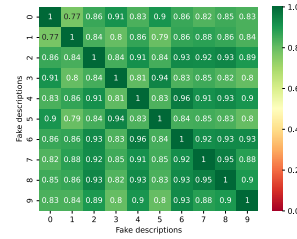
(c) Cosine similarity of the deepfake descriptions for the zero-shot CLIP model



(d) Cosine similarity of the real descriptions for the finetuned *10desc* CLIP model.



(e) Cosine similarity of the finetuned *10desc* CLIP network, between genuine descriptions on the x axis and deepfake ones on the y axis.



(f) Cosine similarity of the fake descriptions for the finetuned *10desc* CLIP model.

Figure 7: Cosine similarities of the embeddings of the real and fake descriptions. The range of the values is between (0,1), with higher values meaning higher similarity.

Method	Similarity		
	Real/Real	Real/Fake	Fake/Fake
CLIP (zero-shot - 10desc)	0.83	**0.78**	0.85
CLIP (10desc)	**0.88**	0.74	**0.89**

Table 8: Mean cosine similarity of the embeddings of the descriptions used to finetune the *10desc* configuration. Higher values mean higher overall mean similarity between the compared embeddings.

9 Discussion

9.1 Zero-shot experiments

It is hypothesized that CLIP possesses semantic understanding of deepfakes from text, but the visual features have not been trained with enough data to accurately classify an image as genuine or fake. Another qualitative experiment conducted in the early stages of working with CLIP supports this hypothesis. The experiment involved an image consisting of two parts: an authentic person on the right side and the corresponding deepfake on the left side. Multiple descriptions were provided to the model to confuse it. CLIP consistently chose the deepfake description that best matched the image. However, when each sub-part of the image was queried separately, CLIP failed to recognize which face was the deepfake, incorrectly predicting the real face as a deepfake and the fake face as genuine. The image related to this experiment can be found in Figure 8 of Appendix B.

Additionally, the experiments conducted in Sub-section 8.1 demonstrate that the pre-trained zero-shot model can achieve improved performance when multiple descriptions are used during inference. It is speculated that the image-text pairs representing deepfakes in the pre-training phase may have been focused around descriptions containing synonymous terms to *"deepfake"*, such as *"artificially created image"*, *"spoofing attempt"*, *"manipulated image"*, or similar phrases. Therefore, relying on only one description is insufficient to capture all the relevant knowledge of the model regarding what is *real* and what is *fake*, specifically for face forgery detection.

The findings from the multi-description configuration in Table 3 and Figure 5 align with some of the experiments in Section "Prompt Engineering and Ensembling" of the original CLIP publication [Radford et al., 2021]. The authors discovered that the model performed better when the descriptions were engineered to match the dataset's context. For instance, using the template *"label, a type of food"* yielded better results than *"An image of a label"* for the classes in the Food101 dataset. Similarly, the performance of CLIP in deepfake detection heavily depends on the choice of descriptions. These results can also be seen as analogous to the ensembling experiments in the

original CLIP paper, where multiple zero-shot classifiers with slightly different contextual descriptions were ensembled by averaging the embeddings. In our case, we average the logits of a single model, resulting in a similar performance improvement compared to using a single description.

Discovering the most effective method for prompting CLIP to achieve optimal results, however, remains a challenging task. While the top-k experiments in Figure 12 and Table 12 demonstrate that performance generally improves as more descriptions are used, there are noticeable fluctuations in the results. These fluctuations may be attributed to certain descriptions being inadequate in relation to the information encountered during the pre-training phase. In other words, not all descriptions may accurately capture the relevant knowledge required for deepfake detection, as also demonstrated in the qualitative experiment of Section 8.1. As a result, finding the right amount and quality of descriptions that align with the model's pre-training data is crucial for achieving consistent and improved performance.

9.2 Vision encoder reset experiments

The results of the fine-tuning experiments in Section 8.2 indicate that the choice of text descriptions can greatly influence the features learned by the visual model, and subsequently, the model's performance when fine-tuning. The best-performing configuration is the *carMoto* experiment, even though the first two deepfake configurations could be considered better suited for the task.

This outcome raises the hypothesis that, given the low robustness of the model for deepfake detection, as seen in Section 8.1, it is possible that features related to other descriptions align well with the task, leading to the carMoto configuration's superior performance compared to the 1desc and 550desc configurations in the fine-tuning process.

Moreover, the experiment demonstrates that it is possible to learn robust visual features for downstream tasks even without pre-trained vision weights. This eliminates the constraint of relying on aligned visual and language features from pre-training on image-text pairs. However, it highlights the ne-

cessity of finding appropriate descriptions for the specific downstream task.

Notably, despite the low robustness of the model in deepfake detection, as observed in Section 8.2, fine-tuning with the proposed method yields promising results. All fine-tuned configurations achieve an improvement in average AUC% performance over the zero-shot CLIP experiments of Section 8.1.

In contrast to the experiments in Section 8.1, the 1desc configuration performs slightly better than the 550desc configuration when fine-tuned. This suggests that in the fine-tuning process, a single description may capture relevant knowledge more effectively than using a larger set of descriptions, for datasets with a limited number of classes.

In summary, the results of these fine-tuning experiments highlight the significant impact of text descriptions on the learned visual features and overall performance. Additionally, the unexpected outcome of the *carMoto* experiment showcases that alignment of features related to different descriptions, unrelated to deepfake detection have the potential to guide the performance of the fine-tuning process, although a way to identify the optimal descriptions for a downstream classification task is needed.

9.3 State-Of-The-Art comparison

The experiments of Section 8.3 yielded some highly encouraging results for the Out-Of-Distribution datasets, with CLIP achieving State-Of-The-Art performance that is on par with the achievements of previous works such as [Shiohara et al., 2022, Guan et al., 2022, Li et al., 2020b]. Compared to other methodologies, however, CLIP offers additional flexibility especially for real-world data, as it does not rely on specific meta-data information during training.

Although CLIP's In-Distribution performance is slightly below the average compared to other methods, in all the fine-tuned configurations, this can possibly be attributed to the limited hyper-parameter tuning. The In-Distribution performance of the *1desc* configuration is also reported in Table 13 of Appendix B. The results demonstrate a fairly balanced performance across the different manipulation techniques, with the *NeuralTextures* being the most

challenging. It is important to note however, that the investigation of In-Distribution performance was not extensively pursued in this work, as our primary focus was the generalization performance of the method on data distributions different from the training set.

In addition to the observations made in Section 8.2, the table highlights that the "1desc" configuration performs better than the configurations with multiple descriptions. This finding suggests that it is easier and more efficient for the network to align all features towards a single description, given our fine-tuning method, for the classification task.

In addition, the "carMoto" configuration trained in Section 8.2 is able to attain similar results to the "1desc" configuration while simplifying the training process. This configuration eliminates the requirement for tuning the text encoder, which can be a resource-intensive task. Additionally, it hypothetically removes the reliance on pre-trained and aligned visual features from the vision encoder, simplifying the requirements of the fine-tuning method. This could allow us to train from scratch visual encoders, for downstream tasks, by aligning them to large language models, pre-trained on billions of data samples, possibly leading to even better generalization. This however is only a hypothesis and would require validation by trying to align arbitrary pre-trained text encoders to image encoders, by solely adjusting the weights of the image encoder.

It is important to note that the comparisons made in Section 8.3 are based on the results reported in the original publications of each method. This approach was chosen for the following two reasons. Firstly, the field of deepfake detection encompasses a wide range of methods, each with its own specific data splitting and preprocessing requirements, making it impractical to validate and test all of them. Secondly, it is assumed that the authors of each method made thorough experimentation to optimize their algorithms and provide reliable results. In addition, careful consideration was given to ensure a fair comparison with our proposed method. The compared methods were selected based on the highest reported results that were compatible with our experimental setup of training on the FaceForensics++ (HQ) dataset and evaluating on DFDC and Celeb-DF datasets, as it was the most common

experiment observed in other works. In the literature, the number of samples used for the evaluation was found to range from 32 to 110 frames per video (estimated at around 20,000 to 60,000 images per dataset). As the value used is mainly arbitrary and can also depend on the method, the threshold of 20,000 images per dataset was chosen as no significant difference in performance was noticed over using more frames.

In response to concerns about the fairness of our results, it is worth considering the possibility of CLIP being pre-trained on similar data distributions as those used in our evaluations. While this is a valid point, it is important to note that the learned features of CLIP alone are insufficient without leveraging them effectively through appropriate descriptions. As demonstrated from the zero-shot experiments, achieving good results requires careful prompt engineering, a task that can quickly become overwhelming and challenging, due to the multitude of concepts and terms, around deepfake detection, that need to be considered. Therefore, alternative approaches, such as the fine-tuning method proposed in this work, are necessary to effectively utilize the pre-trained knowledge of CLIP in a specific downstream task.

9.4 Similarity of embeddings

The experiments conducted in Section 8.4 show that the fine-tuning method introduced some subtle distinctions between the embeddings of real and fake descriptions, making them less similar in latent space. On the other hand, the embeddings of real face descriptions were effectively brought closer together in latent space. Similarly, the embeddings associated with deepfake descriptions also exhibited a higher degree of similarity after fine-tuning. The above properties can also be considered an additional indicator that the fine-tuning process is functioning properly.

Although there may be a slight decrease in the similarity between real and fake descriptions after the fine-tuning, it is important to note that the relative difference is minimal, with only a 4% change. This indicates that the embeddings of real and fake descriptions remain relatively close in the latent space even after fine-tuning. Furthermore, this observation suggests that the fine-tuning procedure does not significantly alter the relative similarities be-

tween the embeddings. The underlying relationships among the descriptions are largely preserved, indicating the robustness of the fine-tuning process in maintaining the overall structure of the embedding space.

However, this observed behavior poses a challenge in our case, as ideally, we would prefer the features corresponding to authentic faces and deepfakes to be further apart in the latent space. This separation could help minimize any overlap between the two classes, leading to improved classification performance. One of the main factors contributing to this behavior is the similarity between the features learned from the visual encoder for real and forged images. Due to the highly overlapping distributions of the two classes, the visual encoder tends to produce similar representations for both types of images. Additionally, since the text encoder is trained to generate embeddings similar to those of the visual encoder, this further contributes to pushing the descriptions of our supposedly "different" classes closer together in the latent space.

To mitigate this issue, one could make use of contrastive learning training paradigms, which have shown great success in improving the separation among classes in latent space, therefore leading to improved performance. The works of [Dong and Shen, 2018, Sun et al., 2020] can be considered some notable contributions in the field that could be adapted into the CLIP fine-tuning method.

10 Limitations & Future work

Although CLIP has demonstrated great potential in generalizing deepfake detection, several limitations have been identified in the current approach. Resolving these limitations could lead to performance improvements and enhance the robustness of face forgery detection.

One limitation pertains to the sampling of descriptions, both for the zero-shot classification task and for the fine-tuning procedure. The experiments conducted in Sub-section 8.1 and 8.2 revealed that different descriptions have varying effects on the classification task of deepfake detection. Optimal sampling of descriptions specific to each downstream task could significantly enhance the interaction with language-vision models and improve overall performance. Furthermore, investigating the properties of the text embeddings that enabled the *carMoto* configuration of Section 8.2 to perform similarly to the jointly trained image and text encoders of Section 8.3 could facilitate the fine-tuning of language-vision networks and reduce computational requirements.

Additionally, the descriptions used in the *10desc* and *550desc* configurations aimed to capture the main distinction between real and fake images without addressing specific characteristics or artifacts found in each individual image. A more effective approach, resembling the pre-training method used in CLIP, would involve including image-text samples with highly informative descriptions of why an image is a deepfake. These descriptions could address attributes such as artifacts or the specific generation technique employed. By combining text descriptions of facial features that CLIP already understands well with motivations for identifying fakes, the model could potentially learn more robust features for discriminating between real and fake images. This method would allow the use of multiple image-text samples during training and eliminate collisions, aligning with the original pre-training methodology.

Moreover, training CLIP to discriminate not only between real and deepfake faces but also between real and artificially generated images more broadly could be beneficial to improve generalization. This approach would aim to develop robust features capable of recognizing fake images and artifacts,

independent of the subject represented.

Given the observed similarity in embeddings between real and fake images, which can be attributed to the considerable overlap between the data distributions of the two classes, it could be beneficial to enhance the separation between the two classes in latent space. One potential approach to achieve this would be to incorporate a contrastive loss term into the training process, aiming to maximize the dissimilarity between embeddings of real and fake image pairs. Contrastive learning has already demonstrated its effectiveness in the domain of deepfake detection, as explored in the work of [Sun et al., 2022].

Another method that is effective for improving the performance of machine learning methods is ensemble learning, introduced in machine learning by the work of [Breiman, 1996]. In the context of CLIP, ensemble learning could involve using multiple fine-tuned models and averaging the logits for real and fake classes. Additionally, exploring the impact of ensembling when multiple CLIP models are fine-tuned solely on the vision encoder with different real-fake descriptions could yield valuable insights. The utilization of ensembling was also found to improve performance in the zero-shot setting in the work of [Radford et al., 2021].

Finally, as CLIP does not require any specific training setup, it could be combined with other deepfake detection methods such as [Shiohara et al., 2022].

11 Conclusions

The aim of this research was to study the impact of vision-language pre-training for the task of face forgery detection and compare the performance to other SOTA methods that mainly rely on visual features. Overall, the conclusion can be drawn that language-vision pre-trained networks can have a beneficial impact in the field of face forgery detection.

In summary, the following contributions are made by our research:

- Firstly, we demonstrate that CLIP can achieve comparable generalization performance to CNN-based models fine-tuned specifically to the task of deepfake detection, by querying the model with various descriptions, without requiring additional optimizations or fine-tuning processes.
- Secondly, we introduce a fine-tuning paradigm for language-vision models, similar to CLIP, that is well-suited for datasets with a limited number of classes. This paradigm addresses the problem of collisions during training that may arise due to the lack of unique image-text pairs.
- Additionally, we achieve State-Of-The-Art results in the task of deepfake detection when pairing our method with a pre-trained CLIP model, demonstrating its effectiveness and competitive performance.
- Finally, we suggest that robust text features have the potential to significantly influence the overall performance of a fine-tuned vision model. Potentially, with adequate descriptions, any pre-trained text encoder model can be used to fine-tune a visual encoder and achieve State-Of-The-Art performance in downstream tasks.

Going into more detail, the generalization performance observed in the original publication of CLIP in the work of [Radford et al., 2021], was similarly translated in the deepfake detection task. As the authors of CLIP argue, CLIP can match the performance of strong fully-supervised baselines in a zero-shot setting, however, the performance is well below the overall SOTA performance in most downstream tasks. In the same way, the zero-shot generalization performance of CLIP presented in Table 3 is able to match that

of the baseline Xception network.

Additionally, the *550desc* experiment, together with the top-k one in Section 8.1, show that it is possible to gain additional performance in the zero-shot setting, by using multiple descriptions of the same class. The reason behind this behavior could be that during the pre-training phase, the model encountered deepfakes, but in text-image pairs with different kinds of descriptions, possibly mentioning *manipulation*, *artificially generated images* or similar terms instead of the specific term *deepfake*. Therefore, one simple description around deepfakes is not enough to encompass all the features relevant to non-genuine images, thus explaining the performance gain when using multiple descriptions focusing on multiple terms around the concept of face forgery. This behavior is explained in the original [Radford et al., 2021] where it is stated that CLIP can struggle with polysemy (words that can have different meanings in different contexts) and synonyms (different words that have similar meanings) and employ ensembling techniques to mitigate the issue.

From the above observations, it is therefore clear that the original CLIP model was exposed to instances of deepfake image-text pairs in the pre-training phase and that there is semantic understanding around the concept of *face forgery*. However, the overall zero-shot performance is well below the SOTA methods presented in Table 6, suggesting that the features are not robust enough to correctly classify most images. Nevertheless, we show that it is possible to fine-tune CLIP to datasets with a limited number of classes, without depending on unique image-text pairs, with a novel adaptation of the original pre-training algorithm from [Radford et al., 2021] that overcomes the problem of collision, and can use multiple descriptions for each class.

The fine-tuned configurations seen in Section 6 managed to train successfully, achieving SOTA results while relaxing the number of constrictions required for training, compared to other SOTA face forgery approaches. Our best-trained model achieves an average AUC% score of 88.15%, while the best overall approach to the best of our knowledge, achieves an average AUC% score of 89.7%. This method of fine-tuning seems promising since generalization under distribution shifts (in this case the DFDC and Celeb-DF evaluation

datasets) was not harmed, but rather improved significantly. The fact that the *1desc* and the *550desc* configurations presented in Sub-section 8.3 provided better results than those in Sub-section 8.2 with frozen text encoder weights, further supports that further investigation is required to understand the limitations of this fine-tuning method compared to only fine-tuning the visual encoder of CLIP models as commonly done. It would be interesting to benchmark the proposed approach on common datasets such as ImageNet to compare it with the current SOTA.

In addition, we demonstrate that it is also possible to achieve SOTA results only by training the image encoder, as seen from the *carMoto* experiment in Section 8.2. Although the reason behind this behavior is not completely clear, the features of the descriptions used in the *carMoto* experiments, "The image of a car." used for genuine images and "The image of a motorcycle." for deepfake images, appear to be more suited than the features extracted from the descriptions of deepfakes and genuine face images, used in the *1desc* and *550desc* configurations of the same section. Nevertheless, two things can be deduced from this experiment.

Firstly, robust enough text features can potentially guide the overall performance of the fine-tuned model, and given performance will depend on the text features used, for the specific downstream task. Secondly, we mention in Section 8.2 that to reach SOTA results it is not required to have a pre-trained visual encoder that can produce similar features to the text encoder. This statement suggests that it would be possible to take any text encoder and use it to train a visual encoder from scratch to produce similar embeddings with those of the text encoder, for a given downstream task, and be able to reach SOTA generalization performance regardless. The approach would be also similar to the work of [Jia et al., 2021], where the authors experimented with aligning various image and text encoders with a noisy image-description pair dataset, of over one billion samples, similar to how it is done in [Radford et al., 2021], and benchmark the zero-shot performance on different downstream tasks. Also, it must be noted that this approach would require a method to find suitable descriptions for the specific downstream task, something that is not addressed in our work.

In addition, an important observation must be acknowledged, regarding training a CLIP model from scratch with our adaptation, as it was not achieved. Our intuition relies on the fact that Transformer models are more effective the more data is used for training. For this reason, trying to train a CLIP network from scratch with such a low amount of text descriptions as in configurations *1desc* and *550desc* is not effective.

As a final note, we want to highlight that employing natural language processing techniques for improving the generalization performance of computer vision tasks is promising. For tasks where mathematical modeling of loss functions that can express the underlying problem is particularly challenging, as in the task of deepfake detection, language could be used as a proxy.

References

[Almars, 2021] Almars, A. (2021). Deepfakes detection techniques using deep learning: A survey. In *Journal of Computer and Communications*, pages 20–35.

[Ba et al., 2016] Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

[Breiman, 1996] Breiman, L. (1996). Bagging predictors. *Machine learning*, 24:123–140.

[Cao et al., 2022] Cao, J., Ma, C., Yao, T., Chen, S., Ding, S., and Yang, X. (2022). End-to-end reconstruction-classification learning for face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4113–4122.

[Chen et al., 2022a] Chen, L., Zhang, Y., Song, Y., Liu, L., and Wang, J. (2022a). Self-supervised learning of adversarial example: Towards good generalizations for deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18710–18719.

[Chen et al., 2022b] Chen, L., Zhang, Y., Song, Y., Wang, J., and Liu, L. (2022b). Ost: Improving generalization of deepfake detection via one-shot test-time training. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems (NeurIPS)*.

[Chen et al., 2020a] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020a). A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607.

[Chen et al., 2020b] Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. E. (2020b). Big self-supervised models are strong semi-supervised learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:22243–22255.

[Chollet, 2017] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1251–1258.

[Dai and Le, 2015] Dai, A. M. and Le, Q. V. (2015). Semi-supervised sequence learning. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28. Curran Associates, Inc.

[Dale et al., 2011] Dale, K., Sunkavalli, K., Johnson, M. K., Vlasic, D., Matusik, W., and Pfister, H. (2011). Video face replacement. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, pages 1–10.

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255.

[Desai and Johnson, 2021] Desai, K. and Johnson, J. (2021). Virtex: Learning visual representations from textual annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11162–11173.

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[Dolhansky et al., 2020] Dolhansky, B., Bitton, J., Pflaum, B., Lu, J., Howes, R., Wang, M., and Ferrer, C. C. (2020). The deepfake detection challenge (dfdc) dataset. *arXiv preprint arXiv:2006.07397*.

[Dolhansky et al., 2019] Dolhansky, B., Howes, R., Pflaum, B., Baram, N., and Ferrer, C. C. (2019). The deepfake detection challenge (dfdc) preview dataset. *arXiv preprint arXiv:1910.08854*.

[Dong et al., 2023] Dong, F., Zou, X., Wang, J., and Liu, X. (2023). Contrastive learning-based general deepfake detection with multi-scale rgb fre-

quency clues. *Journal of King Saud University-Computer and Information Sciences*, pages 90–99.

[Dong et al., 2022] Dong, X., Bao, J., Zhang, T., Chen, D., Gu, S., Zhang, W., Yuan, L., Chen, D., Wen, F., and Yu, N. (2022). Clip itself is a strong fine-tuner: Achieving 85.7% and 88.0% top-1 accuracy with vit-b and vit-l on imagenet. *arXiv preprint arXiv:2212.06138*.

[Dong and Shen, 2018] Dong, X. and Shen, J. (2018). Triplet loss in siamese network for object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 459–474.

[Dosovitskiy et al., 2020] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

[Durall et al., 2020] Durall, R., Keuper, M., and Keuper, J. (2020). Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7887–7896.

[Garrido et al., 2014] Garrido, P., Valgaerts, L., Rehmsen, O., Thormahlen, T., Perez, P., and Theobalt, C. (2014). Automatic face reenactment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4217–4224.

[Goyal et al., 2023] Goyal, S., Kumar, A., Garg, S., Kolter, Z., and Raghunathan, A. (2023). Finetune like you pretrain: Improved finetuning of zero-shot vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19338–19347.

[Grill et al., 2020] Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. (2020). Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:21271–21284.

[Guan et al., 2022] Guan, J., Zhou, H., Hong, Z., Ding, E., Wang, J., Quan, C., and Zhao, Y. (2022). Delving into sequential patches for deepfake detection. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems (NeurIPS)*.

[Haliassos et al., 2021] Haliassos, A., Vougioukas, K., Petridis, S., and Pantic, M. (2021). Lips don't lie: A generalisable and robust approach to face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5039–5049.

[Han et al., 2021] Han, J., Shoeiby, M., Petersson, L., and Armin, M. A. (2021). Dual contrastive learning for unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 746–755.

[Hancock and Bailenson, 2021] Hancock, J. T. and Bailenson, J. N. (2021). The social impact of deepfakes. *Cyberpsychology, Behavior, and Social Networking*, 24(3):149–152.

[He et al., 2020] He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9729–9738.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

[Henaff, 2020] Henaff, O. (2020). Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pages 4182–4192.

[Howard and Ruder, 2018] Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

[Jia et al., 2021] Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., and Duerig, T. (2021). Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916.

[Jia et al., 2022] Jia, J., Chen, X., Yang, A., He, Q., Dai, P., and Liu, M. (2022). Link of transformers in cv and nlp: A brief survey. In *2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*, pages 735–743.

[Joulin et al., 2016] Joulin, A., Van Der Maaten, L., Jabri, A., and Vasilache, N. (2016). Learning visual features from large weakly supervised data. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 67–84.

[Karras et al., 2019a] Karras, T., Laine, S., and Aila, T. (2019a). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4401–4410.

[Karras et al., 2019b] Karras, T., Laine, S., and Aila, T. (2019b). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4401–4410.

[Khodabakhsh et al., 2020] Khodabakhsh, A., Busch, and Christoph (2020). A generalizable deepfake detector based on neural conditional distribution modelling. In *2020 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–5.

[Khosla et al., 2020] Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. (2020). Supervised contrastive learning. *Advances in Neural Information Processing Systems (NeurIPS) (NeurIPS)*, 33:18661–18673.

[Kim et al., 2021] Kim, M., Tariq, S., and Woo, S. S. (2021). Fretal: Generalizing deepfake detection using knowledge distillation and representation

learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1001–1012.

[King, 2009] King, D. E. (2009). Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10(60):1755–1758.

[Köpf et al., 2023] Köpf, A., Kilcher, Y., von Rütte, D., Anagnostidis, S., Tam, Z.-R., Stevens, K., Barhoum, A., Duc, N. M., Stanley, O., Nagyfi, R., et al. (2023). Openassistant conversations–democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*.

[Li et al., 2017] Li, A., Jabri, A., Joulin, A., and Van Der Maaten, L. (2017). Learning visual n-grams from web data. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pages 4183–4192.

[Li et al., 2020a] Li, G., Duan, N., Fang, Y., Gong, M., and Jiang, D. (2020a). Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11336–11344.

[Li et al., 2020b] Li, L., Bao, J., Zhang, T., Yang, H., Chen, D., Wen, F., and Guo, B. (2020b). Face x-ray for more general face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5001–5010.

[Li et al., 2020c] Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., et al. (2020c). Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 121–137.

[Li et al., 2018] Li, Y., Chang, M.-C., and Lyu, S. (2018). In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7.

[Li et al., 2020d] Li, Y., Yang, X., Sun, P., Qi, H., and Lyu, S. (2020d). Celeb-df: A large-scale challenging dataset for deepfake forensics. In *Pro-*

ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

[Liu et al., 2021] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pages 10012–10022.

[Loshchilov and Hutter, 2017] Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

[Mahajan et al., 2018] Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., and Van Der Maaten, L. (2018). Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 181–196.

[McCann et al., 2018] McCann, B., Keskar, N. S., Xiong, C., and Socher, R. (2018). The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

[Mori et al., 1999] Mori, Y., Takahashi, H., and Oka, R. (1999). Image-to-word transformation based on dividing and vector quantizing images with words. In *First International Workshop on Multimedia Intelligent Storage and Retrieval Management (MISRM)*, pages 1–9.

[Nirkin et al., 2019] Nirkin, Y., Keller, Y., and Hassner, T. (2019). Fsgan: Subject agnostic face swapping and reenactment. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pages 7184–7193.

[Perov et al., 2020] Perov, I., Gao, D., Chervoniy, N., Liu, K., Marangonda, S., Umé, C., Dpfks, M., Facenheim, C. S., RP, L., Jiang, J., et al. (2020). Deepfacelab: Integrated, flexible and extensible face-swapping framework. *arXiv preprint arXiv:2005.05535*.

[Peters et al., 2018] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 2227–2237.

[Qian et al., 2020] Qian, Y., Yin, G., Sheng, L., Chen, Z., and Shao, J. (2020). Thinking in frequency: Face forgery detection by mining frequency-aware clues. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M., editors, *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 86–103.

[Quattoni et al., 2007] Quattoni, A., Collins, M., and Darrell, T. (2007). Learning visual representations using images with captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8.

[Radford et al., 2021] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763.

[Radford et al., 2019] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

[Raffel et al., 2020] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

[Rossler et al., 2019] Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., and Niessner, M. (2019). Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pages 1–11.

[Shiohara et al., 2022] Shiohara, Kaede, and Yamasaki, T. (2022). Detecting deepfakes with self-blended images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18720–18729.

[Srivastava et al., 2017] Srivastava, S., Labutov, I., and Mitchell, T. (2017). Joint concept learning and semantic parsing from natural language explanations. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 1527–1536.

[Stiennon et al., 2020] Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. (2020). Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 3008–3021.

[Sun et al., 2021] Sun, K., Liu, H., Ye, Q., Gao, Y., Liu, J., Shao, L., and Ji, R. (2021). Domain general face forgery detection by learning to weight. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2638–2646.

[Sun et al., 2022] Sun, K., Yao, T., Chen, S., Ding, S., Li, J., and Ji, R. (2022). Dual contrastive learning for general face forgery detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2316–2324.

[Sun et al., 2020] Sun, Y., Cheng, C., Zhang, Y., Zhang, C., Zheng, L., Wang, Z., and Wei, Y. (2020). Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Tariq et al., 2019] Tariq, S., Lee, S., Kim, H., Shin, Y., and Woo, S. S. (2019). Gan is a friend or foe? a framework to detect various fake face images. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 1296—-1303.

[Tariq et al., 2021] Tariq, S., Lee, S., and Woo, S. (2021). One detector to rule them all: Towards a general deepfake attack detection framework. In *Proceedings of the Web Conference*, pages 3625–3637.

[Tariq et al., 2020] Tariq, S., Lee, S., and Woo, S. S. (2020). A convolutional lstm based residual network for deepfake video detection. *arXiv preprint arXiv:2009.07480*.

[Thies et al., 2019] Thies, J., Zollhöfer, M., and Nießner, M. (2019). Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, pages 1–12.

[Thies et al., 2016] Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., and Niessner, M. (2016). Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2387–2395.

[Thomee et al., 2016] Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., and Li, L.-J. (2016). Yfcc100m: The new data in multimedia research. *Proceedings of the ACM International Conference on Multimedia*, pages 64—73.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS) (NeurIPS)*, 30.

[Wang and Deng, 2021] Wang, C. and Deng, W. (2021). Representative forgery mining for fake face detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14923–14932.

[Wang et al., 2020] Wang, S.-Y., Wang, O., Zhang, R., Owens, A., and Efros, A. A. (2020). Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8695–8704.

[Wang et al., 2022] Wang, Z., Codella, N., Chen, Y.-C., Zhou, L., Yang, J., Dai, X., Xiao, B., You, H., Chang, S.-F., and Yuan, L. (2022). Clip-td: Clip targeted distillation for vision-language tasks. *arXiv preprint arXiv:2201.05729*.

[Wojewidka, 2020] Wojewidka, J. (2020). The deepfake threat to face biometrics. *Biometric Technology Today*, 2020(2):5–7.

[Wortsman et al., 2022] Wortsman, M., Ilharco, G., Kim, J. W., Li, M., Kornblith, S., Roelofs, R., Lopes, R. G., Hajishirzi, H., Farhadi, A., Namkoong, H., et al. (2022). Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7959–7971.

[Wæver and Buzan, 2020] Wæver, O. and Buzan, B. (2020). Racism and responsibility – the critical limits of deepfake methodology in security studies: A reply to howell and richter-montpetit. *Security Dialogue*, 51(4):386–394.

[Yang and Lim, 2020] Yang, C. and Lim, S.-N. (2020). One-shot domain adaptation for face generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5921–5930.

[Yang et al., 2019] Yang, X., Li, Y., and Lyu, S. (2019). Exposing deep fakes using inconsistent head poses. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8261–8265.

[Zhang et al., 2019] Zhang, X., Karaman, S., and Chang, S.-F. (2019). Detecting and simulating artifacts in gan fake images. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6.

[Zhang et al., 2022] Zhang, Y., Jiang, H., Miura, Y., Manning, C. D., and Langlotz, C. P. (2022). Contrastive learning of medical visual representations from paired images and text. In *Machine Learning for Healthcare Conference*, pages 2–25.

[Zhao et al., 2021] Zhao, H., Zhou, W., Chen, D., Wei, T., Zhang, W., and Yu, N. (2021). Multi-attentional deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2185–2194.

[Zhu et al., 2021] Zhu, X., Wang, H., Fei, H., Lei, Z., and Li, S. Z. (2021). Face forgery detection by 3d decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2929–2939.

[Zhuang et al., 2022] Zhuang, W., Chu, Q., Tan, Z., Liu, Q., Yuan, H., Miao, C., Luo, Z., and Yu, N. (2022). Uia-vit: Unsupervised inconsistency-aware method based on vision transformer for face forgery detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 391–407.

A Datasets

The following appendix presents complementary information about the datasets used to train our models. The prompts used to sample the descriptions of configurations *550desc* and *10desc* from ChatGPT have been reported in Table 9. The prompts were used in succession in the same chat and the outputs were manually inspected and adjusted wherever required. In Table 10 the sub-sample of the descriptions used for the *10desc* configuration trained in Section 8.3 can be found.

B Additional results

This appendix provides supplementary material and additional findings relative to the experiments in the main document. In Figure 8 we find the image relative to the preliminary experiment described in Section 8.1, showing that CLIP has knowledge of deepfakes. Table 12 outlines the exact values of the top-k experiment, showing that using more descriptions when using CLIP is beneficial in a zero-shot setting. In Table 13 the In-Distribution results of configuration *1desc* trained in Sub-Section 8.3 are reported. In the Table, the AUC and precision percentage scores are reported, showing that performance is fairly consistent around the manipulation techniques trained on, with the NeuralTextures dataset being the most challenging.

Prompt to sample 550 deepfake descriptions
Create descriptions for different deepfake / modified or artificially created images. The descriptions should look like this: • "The image exhibits inconsistencies in facial features, hinting at potential deepfake manipulation." • "The image shows signs of facial manipulation, giving the impression of an artificially altered appearance." • "The image has been generated artificially" • "The face in the image has been modified" • "The image has been modified to hide the original face in the image" • "The image is a spoofing attempt to bypass face forgery recognition" • "The image has small inconsistencies in the face, hinting that it has been modified to hide the original identity of the person." The prompts need to be applicable to any image. It is recommended to keep them a bit vague and not go into much detail on why a specific image might be fake. That means no assumptions that artifacts are in a specific place in the face or similar. Create 550 of these descriptions.
Prompt to sample 550 genuine descriptions
Now create 550 similar descriptions for real images that don't have any signs of modifications or of being deepfakes.

Table 9: Prompts to sample deepfake and genuine descriptions from ChatGPT. The two prompts are used in succession in the same chat as the second prompt depends on the description of the first one.

Descriptions used for genuine non-manipulated images
1. The image of an unaltered, authentic person.
2. The picture shows a real face that has not been modified or altered.
3. An image of a genuine person with no inconsistencies around the face.
4. A photograph that shows no signs of digital manipulation, reflecting its authenticity.
5. The image captures a natural and unaltered representation of the person's face.
6. An image that shows no signs of digital tampering, editing, or manipulation, reflecting its authenticity.
7. A photo of a genuine person with no indications of synthetic changes or unnatural adjustments.
8. A picture of a real person where the identity has not been altered.
9. The image shows a person's unaltered face, reflecting their true identity.
10. Facial features in the image exhibit natural proportions and symmetry, indicating no modifications.

Table 10: List of real descriptions used to train the multi-label *10desc* configuration.

Descriptions used for deepfake images
1. There are inconsistencies in the facial details, suggesting the possibility of a deepfake.
2. The person portrayed in the image appears unreal, possibly due to facial modifications.
3. It seems that the image undergoes digital modification, particularly in the face region.
4. Certain aspects of the face in the image appear unnaturally generated, indicating potential deepfake techniques.
5. The image exhibits inconsistencies in the facial structure, implying potential digital manipulation.
6. The facial details in the image raise suspicions of artificial manipulation, possibly through deepfake techniques.
7. The image displays irregularities in the facial proportions, hinting at potential digital manipulation.
8. It seems that the face in the image has been digitally morphed or transformed, indicating potential manipulation.
9. It seems that the face in the image has undergone digital manipulation to resemble someone else.
10. The image presents anomalies in the facial symmetry, casting doubt on the authenticity of the face.

Table 11: List of deepfake descriptions used to train the multi-label *10desc* configuration.

Figure 8: An image example used to understand how much knowledge the CLIP pre-trained model possesses about face forgeries. On the left side, we see the deepfake while on the right side the face of the genuine person. The model always chose the most correct deepfake description when the image was passed as is. However, the model was not able to correctly classify which image was real and which one was a deepfake when the two parts of the image were passed separately. This experiment sustains the hypothesis that CLIP has been pre-trained on some examples of deepfakes, understanding the concept, but not having robust enough visual features to recognize them consistently.

Method	AUC %		
	FF++	CelebDF	DFDC
CLIP (top-3)	50.09	50.04	50.29
CLIP (top-10)	50.22	50.06	50.26
CLIP (top-50)	54.21	49.43	50.98
CLIP (top-100)	61.64	49.01	49.69
CLIP (top-200)	64.01	47.97	52.09
CLIP (top-300)	61.60	49.04	53.41
CLIP (top-400)	60.53	51.28	54.24
CLIP (top-500)	64.51	52.21	55.68
CLIP (top-600)	69.48	52.29	57.24
CLIP (top-700)	72.18	51.81	58.74
CLIP (top-800)	75.06	51.31	61.09
CLIP (top-900)	**78.37**	52.76	64.25
CLIP (top-1000)	77.16	**53.18**	**64.69**
CLIP (top-1100)	78.11	52.13	62.29

Table 12: Benchmark experiments on the zero-shot capabilities of CLIP, for 550 descriptions with top-k sampling.

Method	DF		FS		F2F		NT	
	AUC	Prec	AUC	Prec	AUC	Prec	AUC	Prec
CLIP (1desc)	98.59	92.69	98.42	92.40	98.76	93.37	97.15	90.62

Table 13: In-Distribution results of the *1desc* configuration for each specific manipulation technique. The columns correspond to the following manipulation techniques of the FF++ datasets: DF: Deepfakes, FS: FaceSwap, F2F: Face2Face, NT: NeuralTexture. The *Prec* columns represent precision on the given dataset.