ABSTRACT

Title of dissertation: LEVERAGING DEEP GENERATIVE MODELS FOR ESTIMATION AND RECOGNITION

Koutilya PNVR
Doctor of Philosophy, 2023

Dissertation directed by: Professor David W. Jacobs
Department of Electrical and Computer Engineering

Generative models are a class of statistical models that estimate the joint probability distribution on a given observed variable and a target variable. In computer vision, generative models are typically used to model the joint probability distribution of a set of real image samples assumed to be on a complex high-dimensional image manifold. The recently proposed deep generative architectures such as Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), and diffusion models (DMs) were shown to generate photo-realistic images of human faces and other objects. These generative models also became popular for other generative tasks such as image editing, text-to-image, etc. As appealing as the perceptual quality of the generated images has become, the use of generative models for discriminative tasks such as visual recognition or geometry estimation has not been well studied. Moreover, with different kinds of powerful generative models getting popular lately, it's important to study their significance in other areas of computer vision. In this dissertation, we demon-

strate the advantages of using generative models for applications that go beyond just photo-realistic image generation: Unsupervised Domain Adaptation (UDA) between synthetic and real datasets for geometry estimation; Text-based image segmentation for recognition.

In the first half of the dissertation, we propose a novel generative-based UDA method for combining synthetic and real images when training networks to determine geometric information from a single image. Specifically, we use a GAN model to map both synthetic and real domains into a shared image space by translating just the domain-specific task-related information from respective domains. This is connected to a primary network for end-to-end training. Ideally, this results in images from two domains that present shared information to the primary network. Compared to previous approaches, we demonstrate an improved domain gap reduction and much better generalization between synthetic and real data for geometry estimation tasks such as monocular depth estimation and face normal estimation.

In the second half of the dissertation, we showcase the power of a recent class of generative models for improving an important recognition task: text-based image segmentation. Specifically, large-scale pre-training tasks like image classification, captioning, or self-supervised techniques do not incentivize learning the semantic boundaries of objects. However, recent generative foundation models built using text-based latent diffusion techniques may learn semantic boundaries. This is because they must synthesize intricate details about all objects in an image based on a text description. Therefore, we present a tech-

nique for segmenting real and AI-generated images using latent diffusion models (LDMs) trained on internet-scale datasets. First, we show that the latent space of LDMs (z-space) is a better input representation compared to other feature representations like RGB images or CLIP encodings for text-based image segmentation. By training the segmentation models on the latent z-space, which creates a compressed representation across several domains like different forms of art, cartoons, illustrations, and photographs, we are also able to bridge the domain gap between real and AI-generated images. We show that the internal features of LDMs contain rich semantic information and present a technique in the form of LD-ZNet to further boost the performance of text-based segmentation. Overall, we show up to 6% improvement over standard baselines for text-to-image segmentation on natural images. For AI-generated imagery, we show close to 20% improvement compared to state-of-the-art techniques.

LEVERAGING DEEP GENERATIVE MODELS FOR ESTIMATION AND RECOGNITION

by

Koutilya PNVR

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2023

Advisory Committee:
Professor David W. Jacobs, Chair/Advisor
Professor Joseph Jaja
Professor Behtash Babadi
Professor Jia-Bin Huang
Professor Maria K. Cameron (Dean's representative)

Dedication

To my family —

Subrahmanyam Ponukupati

Sudha Ponukupati

Syamala Pisapati

Sandilya Ponukupati

Sruthi Ponukupati

Vaishnavi Ponukupati

Sindhura Purnima Vempati

For their constant support, love, sacrifice and selflessness.

Acknowledgments

I wish to express my deepest gratitude to the remarkable individuals who have been instrumental in my Ph.D. journey, contributing immeasurably to my growth and success.

First and foremost, I extend my sincere appreciation to my advisor, Prof. David Jacobs. Despite my non-computer-vision background, he offered me the invaluable opportunity to work closely with him. I am profoundly thankful for the numerous research meetings and brainstorming sessions, which not only broadened my research horizons but also nurtured my ability to approach complex problems. His unwavering consideration for my circumstances has left an indelible mark, and I couldn't have asked for a more exceptional Ph.D. advisor.

I am deeply honored to have Prof. Joseph JaJa, Prof. Behtash Babadi, Prof. Jia-Bin Huang, and Prof. Maria K. Cameron as members of my dissertation committee. Their commitment to serving on my committee and providing invaluable feedback to enhance the quality of this dissertation is greatly appreciated.

My gratitude extends to my remarkable mentors, Bharat Singh and Hao Zhou, who have been constant sources of support during the challenging phases of my Ph.D. Their participation in research meetings and continuous motivation to explore fresh perspectives on research problems have been transformative. In particular, Bharat's close collaboration and the research skills he imparted are beyond measure. Their guidance has been pivotal, and I owe a significant portion of my progress to their precious mentorship.

I would like to acknowledge Dr. Varaprasad Bandaru for providing me with opportunities from the early stages of my academic journey, beginning with my master's program. His enduring belief in my capabilities and involvement in his remarkable research projects have been essential in broadening the breadth of my knowledge during my Ph.D. journey.

My fellow research peers at the University of Maryland, including students from the research groups of Prof. David Jacobs, Prof. Abhinav Shrivastava, and Prof. Tom Goldstein, have been a constant source of enlightening discussions and camaraderie.

I am grateful to my colleagues from internships, including Pallabi Ghosh, Behjat Siddiquie from Amazon, and Abhijit Bendale, Pranav Mistry from STAR Labs, for the wonderful opportunities they provided, exposing me to real-world experiences.

My thanks go to the International Student and Scholar Services (ISSS), the graduate school, the staff at the ECE and CS departments, and UMIACS for their friendly, liberal, and supportive approach. I will cherish the memories of my student life and the warmth of the university.

To my friends - Shankar Reddy, Dwith CYN, Pallavi Chirumamilla, Sai Deepika Regani, Anirudh Mothukuri, Likhith Anvhesh, Sriram Vasudevan, Sai Sreedhar Varada, Mounika Chintakayala, Raghuvaran Yaramasu, Avinash Bheemineni, Spandana Gorantla, Harika Vakkanthula, Sreeharsha Vardhan Annu, and Manvitha Sree who have been my pillars of strength, offering relentless support and creating wonderful memories, I extend my heartfelt appreciation. Your

friendships have not only aided my personal growth but have also made my Ph.D. journey exceptionally smooth, making you a cherished part of my family.

I would also like to express my sincere thanks to Sindhura Purnima, who entered my life at a crucial stage, offering constant support and understanding. I wholeheartedly believe she is my lucky charm, bringing much-needed fortune at a precious time.

Last, but certainly not least, I owe a profound debt of gratitude to my parents and family members. Their constant motivation and belief in me, through both the good and challenging times, have been the cornerstone of my journey. I am forever indebted to them for the unwavering support and sacrifices they made to help me reach the point where I stand today.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Motivation

The recently proposed deep generative architectures such as Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs) and Diffusion models (DMs) were shown to exhibit photo-realistic image generation quality. Many generative applications such as image editing, text-to-image etc also became popular with these models. However, the use of these generative models for tasks such as representation learning, visual recognition or geometry estimation has been little explored. Typically, such discriminative tasks are solved with CNN or transformer based classifiers that excel at obtaining decision boundaries between classes in the training data. Deep generative models on the other hand, estimate the joint probability distribution of the entire training data. Such models hold more information about the training data and are capable of generating realistic looking samples from the distribution. Moreover, with the size of the datasets getting bigger and the architectures becoming more powerful, exploring the importance of deep generative models for tasks that go beyond just image generation becomes critical.

Generative models have been studied for tasks such as representation learn-

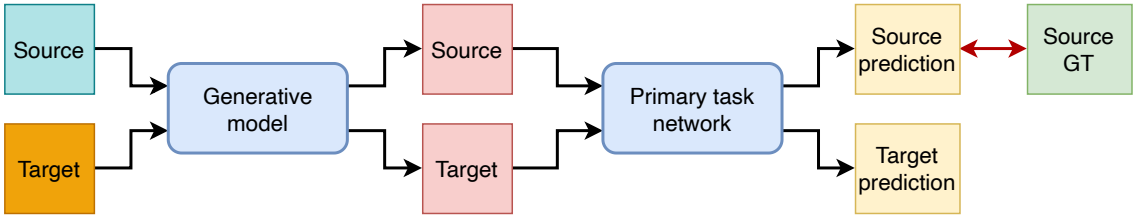


Figure 1.1: Generative models can be used to reduce domain gap between labeled source and unlabeled target domains.

ing [1–7], synthetic data generation [8–10], domain adaptation [11–15] etc. However the rapid progress in the generative models research and the underlying techniques did not scale similarly in these areas. In this dissertation, we attempt to explore and leverage specific deep generative models to improve performance in estimation and recognition tasks namely 1) Unsupervised domain adaptation for geometry estimation and 2) Text-based image segmentation, respectively.

Unsupervised domain adaptation refers to the problem of reducing the domain gap between a labeled source domain and an unlabeled target domain. For geometry estimation such as monocular depth estimation (MDE) and face normal estimation (FNE), some lines of work depend on the vast amount of labeled synthetic data as the source domain and attempt to make it generalize to the real data. Previous works that used generative models for unsupervised geometry estimation, proposed to translate the synthetic data into real-like or vice-versa. However, such an inter-domain mapping is an unnecessarily challenging problem for the generative model and would serve as a bottleneck for the downstream primary task network. We propose a better way to reduce the domain gap by using a GAN based framework that translates just the right amount of information

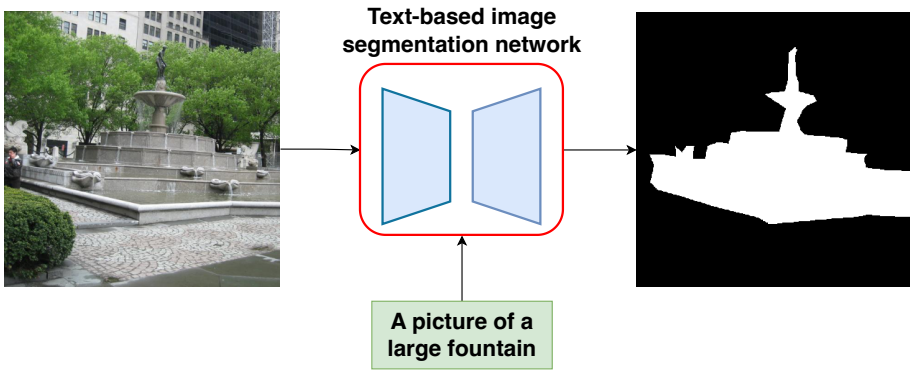


Figure 1.2: Text-based image segmentation aims to segment regions in the image that refer to an input text prompt.

from both synthetic and real domains into a shared image space. A high level overview of the proposed approach is illustrated in Figure 1.1. This shared image space is shown to have better properties in terms of domain generalization for geometry estimation. Specifically, we observe it is only necessary to translate the domain-specific task related information of respective domains into a shared image space. This mapping need not modify the information of original domains that is not related to the primary task as the primary network will learn to ignore them regardless. This simple and intuitive formulation combined with the image translation ability of the generative model, helps the primary task network to look at shared information from both domains with much less domain gap leading to better generalization.

Also, with recent advances in diffusion models (DMs) [16, 17] in unconditional and class conditional settings, they have started gaining more traction compared to GANs. This class of generative models became even more popular for their generated visual quality in text-to-image tasks. Recently, latent diffusion

models (LDMs) [18] were proposed that operate on a perceptually compressed latent space obtained from an internal first stage. LDMs became a popular choice for text-to-image applications for their ability to learn and operate with lower computational cost and on large scale datasets. Such large scale LDMs were shown to exhibit photo-realistic text-to-image visual quality and lead to several visual-linguistic applications such as text guided image inpainting, personalized text-to-image etc. This indicates that pretrained LDMs contain semantic information about various objects from the internet. However, the usefulness of these powerful LDMs have not been explored for text-based recognition problems such as text-based segmentation task illustrated in Figure 1.2. In this dissertation, we propose a text-based segmentation network named LD-ZNet that utilizes an LDM pretrained on large datasets. We show that the segmentation network, with the help of LDM, learns knowledge of novel concepts from the internet without requiring annotations. Overall, our LD-ZNet can segment objects from the internet in various imagery such as real, AI-Generated, animations, illustrations and celebrity images.

1.2 Dissertation Outline and Contributions

1.2.1 Leveraging GANs for Unsupervised Geometry Estimation (Chapter 2)

In this Chapter, we propose a novel generative-based UDA method for combining labeled-synthetic and unlabeled-real images when training networks to

determine geometric information from a single image. Our proposal outlines a strategy to project both image categories into a single, shared domain. This shared domain acts as input to the primary network during end-to-end training. Consequently, the primary network learns from the shared information of both domains and generalizes much better to real-images during test-time. Our experiments demonstrate significant improvements over the state-of-the-art in two important domains, surface normal estimation of human faces and monocular depth estimation for outdoor scenes, both in an unsupervised setting.

1.2.2 Leveraging LDMs for Text-Based Segmentation (Chapter 3)

In this Chapter, we propose LD-ZNet a text-based segmentation network that uses an LDM pretrained on large-scale data. Specifically, we suggest a way to use the z-space and the internal representations inside the LDM to improve segmentation performance for novel concepts on various imagery such as real, AI-generated, animations, illustrations and celebrity images. We additionally create a new dataset named AIGI consisting of AI-Generated images along with object labels and categorical captions for evaluating the generalization ability of text-based segmentation methods to AI-Generated content. We show a huge improvement of around 20% for LD-ZNet over existing text-based segmentation methods on the AIGI dataset.

1.2.3 Bidirectional Convolutional LSTM for the Detection of Violence in Videos (Appendix A)

[1]The field of action recognition has gained tremendous traction in recent years. A subset of this, detection of violent activity in videos, is of great importance, particularly in unmanned surveillance or crowd footage videos. In this appendix, we explore this problem on three standard benchmarks widely used for violence detection: the Hockey Fights, Movies, and Violent Flows datasets. To this end, we introduce a Spatiotemporal Encoder, built on the Bidirectional Convolutional LSTM (BiConvLSTM) architecture. The addition of a bidirectional temporal encoding and the elementwise max pooling of these encodings in the Spatiotemporal Encoder is novel in the field of violence detection. This addition is motivated by a desire to derive better video representations via leveraging long-range information in both temporal directions of the video. We find that the Spatiotemporal network is comparable in performance with existing methods for all of the above datasets. A simplified version of this network, the Spatial Encoder is sufficient to match state-of-the-art performance on the Hockey Fights and Movies datasets. However, on the Violent Flows dataset, the Spatiotemporal Encoder outperforms the Spatial Encoder.

[1]This is placed in the appendix because it is an early thesis work that does not directly connect to the main content of this dissertation.

Chapter 2: GANs for Unsupervised Geometry Estimation

[1]Understanding geometry from images is a fundamental problem in computer vision. It has many important applications. For instance, Monocular Depth Estimation (MDE) is important for synthetic object insertion in computer graphics [20], grasping in robotics [21] and safety in self-driving cars. Face Normal Estimation can help in face image editing applications such as relighting [22–24]. However, it is extremely hard to annotate real data for these regression tasks. Synthetic data and their ground truth labels, on the other hand, are easy to generate and are often used to compensate for the lack of labels in real data. Deep models trained on synthetic data, unfortunately, usually perform poorly on real data due to the domain gap between synthetic and real distributions. To deal with this problem, several research studies [25–28] have proposed unsupervised domain adaptation methods to take advantage of synthetic data by mapping it into the real domain or vice versa, either at the feature level or image level. However, mapping examples from one domain to another domain itself is a challenging problem that can limit performance.

We observe that finding such a mapping solves an unnecessarily difficult

[1]Work done with Hao Zhou and David Jacobs. Accepted [19] in CVPR 2020.

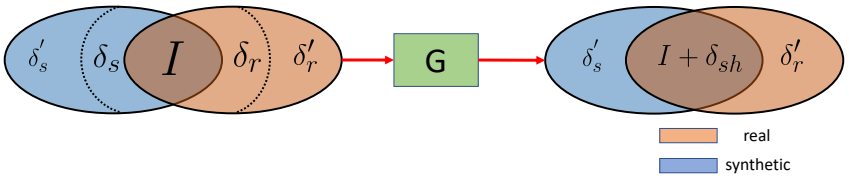


Figure 2.1: We propose to reduce the domain gap between synthetic and real by mapping the corresponding domain specific information related to the primary task (δ_s, δ_r) into shared information δ_{sh}, preserving everything else.

problem. To train a regressor that applies to both real and synthetic domains, it is only necessary that we map both to a new representation that contains the task-relevant information present in both domains, in a common form. The mapping need not alter properties of the original domain that are irrelevant to the task since the regressor will learn to ignore them regardless.

To see this, we consider a simplified model of our problem. We suppose that real and synthetic images are formed by two components: domain agnostic (which has semantic information shared across synthetic and real, and is denoted as I) and domain specific. We further assume that domain specific information has two sub-components: domain specific information unrelated to the primary task (denoted as δ'_s and δ'_r for synthetic and real images respectively) and domain specific information related to the primary task (δ_s, δ_r). So real and synthetic images can be represented as: $x_r = f(I, \delta_r, \delta'_r)$ and $x_s = f(I, \delta_s, \delta'_s)$ respectively.

We believe the domain gap between $\{\delta_s$ and $\delta_r\}$ can affect the training of the primary network, which learns to expect information that is not always present. The domain gap between $\{\delta'_s$ and $\delta'_r\}$, on the other hand, can be bypassed by the

primary network since it does not hold information needed for the primary task. For example, in real face images, information such as the color and texture of the hair is unrelated to the task of estimating face normals but is discriminative enough to distinguish real from synthetic faces. This can be regarded as domain specific information unrelated to the primary task i.e., δ'_r. On the other hand, shadows in the real and synthetic images, due to the limitations of the rendering engine, may have different appearances but may contain depth cues that are related to the primary task of MDE in both domains. The simplest strategy, then, for combining real and synthetic data is to map δ_s and δ_r to a shared representation, δ_{sh}, while not modifying δ'_s and δ'_r as shown in Figure 2.1.

Recent research studies show that a shared network for synthetic and real data can help reduce the discrepancy between images in different domains. For instance, [22] achieved state-of-the-art results in face normal estimation by training a unified network for real and synthetic data. [13] learned the joint distribution of multiple domain images by enforcing a weight-sharing constraint for different generative networks. Inspired by these research studies, we define a unified mapping function G, which is called SharinGAN, to reduce the domain gap between real and synthetic images.

Different from existing research studies, our G is trained so that minimum domain specific information is removed. This is achieved by pre-training G as an auto-encoder on real and synthetic data, i.e., initializing G as an identity function. Then G is trained end-to-end with reconstruction loss in an adversarial framework, along with a network that solves the primary task, further pushing

G to map information relevant to the task to a shared domain.

As a result, a successfully trained G will learn to reduce the domain gap existing in δ_s and δ_r, mapping them into a shared domain δ_{sh}. G will leave I unchanged. δ'_s and δ'_r can be left relatively unchanged when it is difficult to map them to a common representation. Mathematically, $G(x_s) = f(I, \delta_{sh}, \delta'_s)$ and $G(x_r) = f(I, \delta_{sh}, \delta'_r)$. If successful, G will map synthetic and real images to images that may look quite different to the eye, but the primary task network will extract the same information from both.

We apply our method to unsupervised monocular depth estimation using virtual KITTI (vKITTI) [29] and KITTI [30] as synthetic and real datasets respectively. Our method reduces the absolute error in the KITTI eigen test split and the test set of Make3D [31] by 23.77% and 6.45% respectively compared with the state-of-the-art method [27]. Additionally, our proposed method improves over SfSNet [22] on face normal estimation. It yields an accuracy boost of nearly 4.3% for normal prediction within 20° ($Acc < 20°$) of ground truth on the Photoface dataset [32].

2.1 Related Work

Monocular Depth Estimation has long been an active area in computer vision. Because this problem is ill-posed, learning-based methods have predominated in recent years. Many early learning works applied Markov Random Fields (MRF) to infer the depth from a single image by modeling the relation between

nearby regions [31, 33, 34]. These methods, however, are time-consuming during inference and rely on manually defined features, which have limitations in performance.

More recent studies apply deep Convolutional Neural Networks (CNNs) [35–42] to monocular depth estimation. Eigen [35] first proposed a multi-scale deep CNN for depth estimation. Following this work, [36] proposed to apply CNNs to estimate depth, surface normal and semantic labels together. [37] combined deep CNNs with a continuous CRF for monocular depth estimation. One major drawback of these supervised learning-based methods is the requirement for a huge amount of annotated data, which is hard to obtain in reality.

With the emergence of large scale, high-quality synthetic data [29], using synthetic data to train a depth estimator network for real data became popular [26, 27]. The biggest challenge for this task is the large domain gap between synthetic data and real data. [28] proposed to first train a depth prediction network using synthetic data. A style transfer network is then trained to map real images to synthetic images in a cycle consistent manner [43]. [25] proposed to adapt the features of real images to the features of synthetic images by applying adversarial loss on latent features. A content congruent regularization is further proposed to avoid mode collapse. T^2Net [26] trained a network that translates synthetic data into real at the image level and further trained a task network in this translated domain. GASDA [27] proposed to train the network by incorporating epipolar geometry constraints for real data along with the ground truth labels for synthetic data. All these methods try to align two domains by transferring one

domain to another. Unlike these works, we propose a mapping function G, also called SharinGAN, to just align the domain specific information that affects the primary task, resulting in a minimum change in the images in both domains. We show that this makes learning the primary task network much easier and can help it focus on the useful information.

Self-supervised learning is another way to avoid collecting ground truth labels for monocular depth estimation. Such methods need monocular videos [44–47], stereo pairs [48–51], or both [47] for training. Our proposed method is complementary to these self-supervised methods, it does not require this additional data, but can use it when available.

Face Geometry Estimation is a sub-problem of inverse face rendering which is the key for many applications such as face image editing. Conventional face geometry estimation methods are usually based on 3D Morphable Models (3DMM) [52]. Recent studies demonstrate the effectiveness of deep CNNs for solving this problem [22, 53–58]. Thanks to the 3DMM, generating synthetic face images with ground truth geometry is easy. [22,53,54] make use of synthetic face images with ground truth shape to help train a network for predicting face shape using real images. Most of these works initially pre-train the network with synthetic data and then fine-tune it with a mix of real and synthetic data, either using no supervision or weak supervision, overlooking the domain gap between real and synthetic face images. In this work, we show that by reducing the domain gap between real and synthetic data using our proposed method, face geometry can be better estimated.

Domain Adaptation using GANs There are many works [11–15] that use a GAN framework to perform domain adaptation by mapping one domain into another via a supervised translation. However, most of these show performance on just toy datasets in a classification setting. We attempt to map both synthetic and real domains into a new shared domain that is learned during training and use this to solve complex problems of unsupervised geometry estimation. Moreover, we apply adversarial loss at the image level for our regression task, in contrast to some of the above previous works where domain invariant feature engineering sufficed for classification tasks.

2.2 Approach

To compensate for the lack of annotations for real data and to train a primary task network on easily available synthetic data, we propose SharinGAN to reduce the domain gap between synthetic and real. We aim to train a primary task network on a shared domain created by SharinGAN, which learns the mapping function $G : x_r \mapsto x_r^{sh}$ and $G : x_s \mapsto x_s^{sh}$, where $x_k = f(I, \delta_k, \delta'_k)$; $x_k^{sh} = f(I, \delta_{sh}, \delta'_k)$; $k \in \{r, s\}$ as shown in Figure 2.1. G allows the primary task network to train on a shared space that holds the information needed to do the primary task, making the network more applicable to real data during testing.

To achieve this, an adversarial loss is used to find the shared information, δ_{sh}. This is done by minimizing the discrepancy in the distributions of x_r^{sh} and x_s^{sh}. But at the same time, to preserve the domain agnostic information (shared

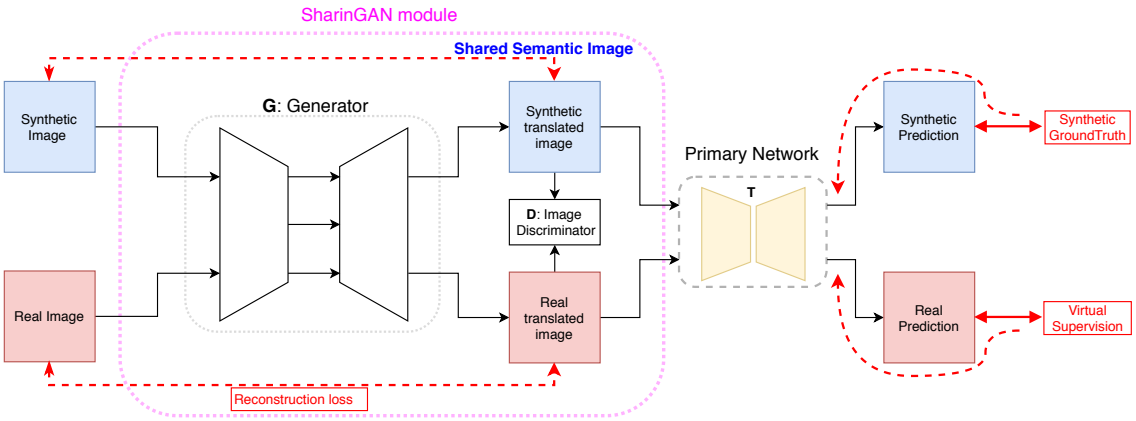


Figure 2.2: Overview of the model architecture. Red dashed arrows indicate the loss computations.

semantic information I), we use reconstruction loss. Now, without a loss from the primary task network, G might change the images so that they don't match the labels. To prevent that, we additionally use a primary task loss for both real and synthetic examples to guide the generator. It is important to note that both the translations from synthetic to real and vice versa are equally crucial for this symmetric setup to find a shared space. To facilitate that, we use a form of weak supervision we call virtual supervision. Some possible virtual supervisions include a prior on the input data or a constraint that can narrow the solution space for the primary task network (details discussed in 2.2.2.2). For synthetic examples, we use the known labels.

Adversarial, Reconstruction and Primary task losses together train the generator and primary task network to align the domain specific information $\{\delta_s, \delta_r\}$ in both the domains into a shared space δ_{sh}, preserving everything else.

2.2.1 Framework

In this work, we propose to train a generative network which is called SharinGAN, to reduce the domain gap between real and synthetic data so as to help to train the primary network. Figure 2.2 shows the framework of our proposed method. It contains a generative network G, a discriminator on image-level D that embodies the SharinGAN module and a task network T to perform the primary task. The generative network G takes either a synthetic image x_s or real image x_r as input and transforms it to x_s^{sh} or x_r^{sh} in an attempt to fool D. Different from existing works that transfer images in one domain to another [26–28], our generative network G tries to map the domain specific parts δ_s and δ_r of synthetic and real images to a shared space δ_{sh}, leaving δ_s' and δ_r' unchanged. As a result, our transformed synthetic and real images (x_s^{sh} and x_r^{sh}) have fewer differences from x_s and x_r. Our task network T then takes the transformed images x_s^{sh} and x_r^{sh} as input and predicts the geometry. The generative network G and task network T are trained together in an end-to-end manner.

2.2.2 Losses

2.2.2.1 Losses for Generative Network

We design a single generative network G for synthetic and real data since sharing weights can help align distributions of different domains [13]. Moreover, existing research studies such as [22, 54] also demonstrate that a unified frame-

work works reasonably well on synthetic and real images. In order to map δ_s and δ_r to a shared space δ_{sh}, we apply adversarial loss [59] at the image level. More specifically, we use the Wasserstein discriminator [60] that uses the Earth-Mover's distance to minimize the discrepancy between the distributions for synthetic and real examples $\{G(x_s), G(x_r)\}$, i.e.:

$$L_W(D,G) = \mathbb{E}_{x_s}[D(G(x_s))] - \mathbb{E}_{x_r}[D(G(x_r))], \tag{2.1}$$

D is a discriminator and G_e is the encoder part of the generator. Following [61], to overcome the problem of vanishing or exploding gradients due to the weight clipping proposed in [60], a gradient penalty term is added for training the discriminator:

$$L_{gp}(D) \quad = \quad (\|\nabla_{\hat{h}} D(\hat{h})\|_2 - 1)^2 \tag{2.2}$$

Our overall adversarial loss is then defined as:

$$L_{adv} \quad = \quad L_W(D,G) - \lambda L_{gp}(D) \tag{2.3}$$

where λ is chosen to be 10 while training the discriminator and 0 while training the generator.

Without any constraints, the adversarial loss may learn to remove all domain specific parts δ and δ' or even some of the domain agnostic part I in order to fool the discriminator. This may lead to loss of geometric information, which can degrade the performance of the primary task network T. To avoid this, we propose to use the self-regularization loss similar to [62] to force the transformed

image to keep as much information as possible:

$$L_r = \|G(x_s) - x_s\|_2^2 + \|G(x_r) - x_r\|_2^2. \tag{2.4}$$

2.2.2.2 Losses for the Task Network

The task network takes transformed synthetic or real images as input and predicts geometric information. Since the ground truth labels for synthetic data are available, we apply a supervised loss using these ground truth labels. For real images, domain specific losses or regularizations are applied as a form of virtual supervision for training according to the task. We apply our proposed SharinGAN to two tasks: monocular depth estimation (MDE) and face normal estimation (FNE). For MDE, we use the combination of depth smoothness and geometric consistency losses used in GASDA [27] as the virtual supervision. For FNE however, for virtual supervision we use the pseudo supervision used in SfS-Net [22]. We use the term "virtual supervision" to summarize these two losses as a kind of weak supervision on the real examples.

2.2.2.3 Monocular Depth Estimation

To make use of ground truth labels for synthetic data, we apply L_1 loss for predicted synthetic depth images:

$$L_1 = \|\hat{y}_s - y_s^*\|_1 \tag{2.5}$$

where $\hat{y}_s$ is the predicted synthetic depth map and y_s^* is its corresponding ground truth. Following [27], we apply smoothness loss on depth L_{DS} to encourage it to

be consistent with local homogeneous regions. Geometric consistency loss L_{GC} is applied so that the task network can learn the physical geometric structure through epipolar constraints. L_{DS} and L_{GC} are defined as:

$$L_{DS} = e^{-\nabla x_r} \|\nabla \hat{y}_r\| \tag{2.6}$$

$$L_{GC} = \eta \frac{1 - SSIM(x_r, x'_{rr})}{2} + \mu \|x_r - x'_{rr}\|, \tag{2.7}$$

$\hat{y}_r$ represents the predicted depth for the real image and ∇ represents the first derivative. x_r is the left image in the KITTI dataset [30]. x'_{rr} is the inverse warped image from the right counterpart of x_r based on the predicted depth $\hat{y}_r$. The KITTI dataset [30] provides the camera focal length and the baseline distance between the cameras. Similar to [27], we set η as 0.85 and μ as 0.15 in our experiments. The overall loss for the task network is defined as:

$$L_T = \beta_1 L_{DS} + \beta_2 L_1 + \beta_3 L_{GC}, \tag{2.8}$$

where $\beta_1 = 0.01, \beta_2 = \beta 3 = 100$.

2.2.2.4 Face Normal Estimation

SfSnet [22] currently achieves the best performance on face normal estimation. We thus follow its setup for face normal estimation and apply "SfS-supervision" for both synthetic and real images during training.

$$L_T = \lambda_{recon} L_{recon} + \lambda_N L_N + \lambda_A L_A + \lambda_{Light} L_{Light}, \tag{2.9}$$

where L_{recon}, L_N and L_A are L_1 losses on the reconstructed image, normal and albedo, whereas L_{light} is the L2 loss over the 27 dimensional spherical harmonic

coefficients. The supervision for real images is from the "pseudo labels", obtained by applying a pre-trained task network on real images. Please refer to [22] for more details.

2.2.2.5 Overall loss

The overall loss used to train our geometry estimation pipeline is then defined as:

$$L = \alpha_1 L_{adv} + \alpha_2 L_r + \alpha_3 L_T. \tag{2.10}$$

where $(\alpha_1, \alpha_2, \alpha_3) = (1, 10, 1)$ for monocular depth estimation task and $(\alpha_1, \alpha_2, \alpha_3) = (1, 10, 0.1)$ for face normal estimation task.

2.3 Experiments

We apply our proposed SharinGAN to monocular depth estimation and face normal estimation. We discuss the details of the experiments in this section.

2.3.1 Monocular Depth Estimation

2.3.1.1 Datasets

Following [27], we use vKITTI [29] and KITTI [30] as synthetic and real datasets to train our network. vKITTI contains 21,260 image-depth pairs, which are all used for training. KITTI [30] provides 42,382 stereo pairs, among which, 22,600 images are used for training and 888 are used for validation as suggested by [27].

2.3.1.2 Implementation details

We use a generator G and a primary task network T, whose architectures are identical to [27]. We pre-train the generative network G on both synthetic and real data using reconstruction loss L_r. This results in an identity mapping that can help G to keep as much of the input image's geometry information as possible. Our task network is pre-trained using synthetic data with supervision. G and T are then trained end to end using Equation 2.10 for 150,000 iterations with a batch size of 2, by using an Adam optimizer with a learning rate of $1e-5$. The best model is selected based on the validation set of KITTI.

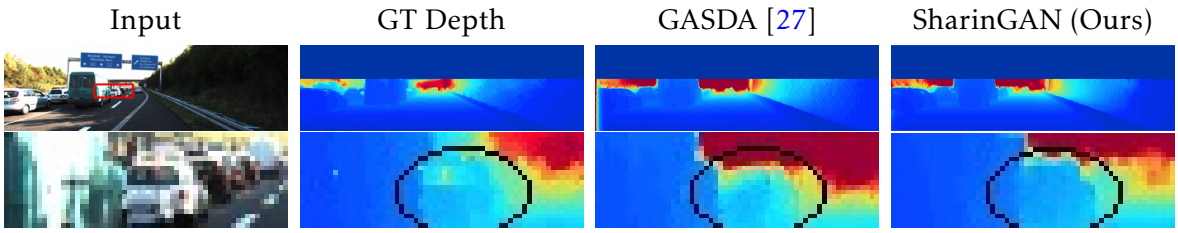
2.3.1.3 Results

Table 2.1 shows the quantitative results on the eigen test split of the KITTI dataset for different methods on the MDE task. The proposed method outperforms the previous unsupervised domain adaptation methods for MDE [26, 27] on almost all the metrics. Especially, compared with [27], we reduce the absolute error by 19.7% and 21.0% on 80m cap and 50m cap settings respectively. Moreover, the performance of our method is much closer to the methods in a supervised setting [35, 37, 63], which was trained on the real KITTI dataset with ground truth depth labels. Figure 2.3 visually compares the predicted depth map from the proposed method with [27]. We show three typical examples: near distance, medium distance, and far distance. It shows that our proposed method performs much better for predicting depth at details. For instance, our predicted

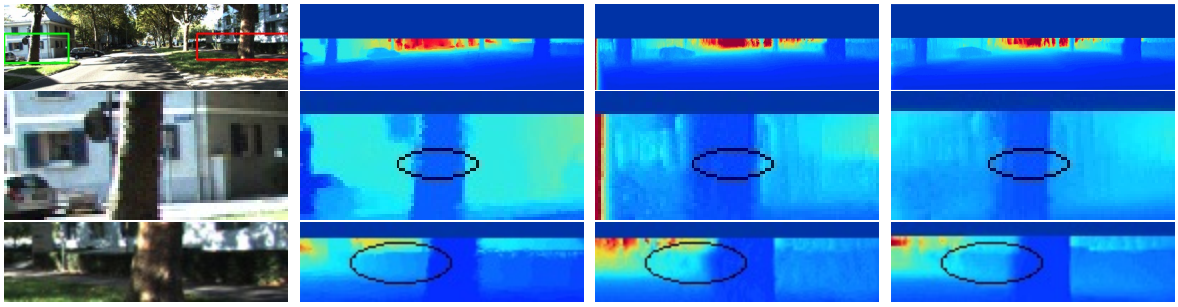
Method	Supervised	Dataset	Cap	Error Metrics, lower is better				Accuracy Metrics, higher is better		
				Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen [35]	Yes	K	80m	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu [37]	Yes	K	80m	0.202	1.614	6.523	0.275	0.678	0.895	0.965
All synthetic (baseline)	No	S	80m	0.253	2.303	6.953	0.328	0.635	0.856	0.937
All real (baseline)	No	K	80m	0.158	1.151	5.285	0.238	0.811	0.934	0.970
GASDA [27]	No	K+S	80m	0.149	1.003	**4.995**	0.227	0.824	0.941	0.973
SharinGAN (proposed)	No	K+S	80m	**0.116**	**0.939**	5.068	**0.203**	**0.850**	**0.948**	**0.978**
Kuznietsov [63]	Yes	K	50m	0.117	0.597	3.531	0.183	0.861	0.964	0.989
Garg [64]	No	K	50m	0.169	1.080	5.104	0.273	0.740	0.904	0.962
Godard [48]	No	K	50m	0.140	0.976	4.471	0.232	0.818	0.931	0.969
All synthetic (baseline)	No	S	50m	0.244	1.771	5.354	0.313	0.647	0.866	0.943
All real (baseline)	No	K	50m	0.151	0.856	4.043	0.227	0.824	0.940	0.973
Kundu [25]	No	K+S	50m	0.203	1.734	6.251	0.284	0.687	0.899	0.958
T2Net [26]	No	K+S	50m	0.168	1.199	4.674	0.243	0.772	0.912	0.966
GASDA [27]	No	K+S	50m	0.143	0.756	3.846	0.217	0.836	0.946	0.976
SharinGAN (proposed)	No	K+S	50m	**0.109**	**0.673**	**3.77**	**0.190**	**0.864**	**0.954**	**0.981**

Table 2.1: MDE Results on eigen test split of KITTI dataset [35] . For the training data, K: KITTI dataset and S: vKITTI dataset. Methods highlighted in light gray, use domain adaptation techniques and the non-highlighted rows correspond to supervised methods.

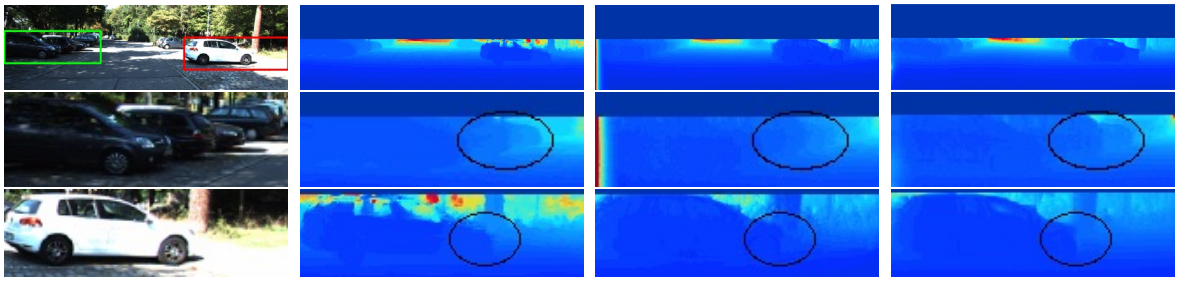
depth map can better preserve the shape of the car (Figure 2.3 (a) and (c)) and the structure of the tree and the building behind it (Figure 2.3 (b)). This shows the advantage of our proposed SharinGAN compared with [27]. [27] learns to transfer real images to the synthetic domain and vice versa, which solves a much harder problem compared with SharinGAN, which removes a minimum of domain specific information. As a result, the quality of the transformation for [27] may not be as good as the proposed method. Moreover, the unsupervised transformation cannot guarantee to keep the geometry information unchanged.



(a) The second row shows the corresponding region in the red box of the first row. The depth of the faraway car is better estimated by SharinGAN than GASDA.

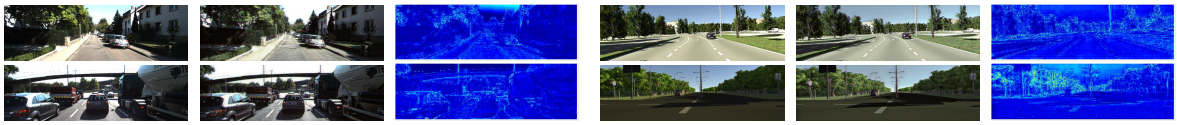


(b) The second and third row shows the corresponding region in the green and red box of the first row. The depth of the tree to the left (green) and shrubs behind the tree in the right are better estimated by SharinGAN.



(c) The second and third row shows the corresponding regions in the green and red boxes of the first row. The boundaries and the depth of the cars are better estimated by SharinGAN.

Figure 2.3: Qualitative comparisons of SharinGAN with GASDA [27]. Ground truth (GT) has been interpolated (and the unavailable top regions are masked out) for visualization purposes. Note that in addition to various other aspects mentioned above, we are also able to remove the boundary artifacts present in the depth maps of GASDA.



(a) x_r (b) $x_r^{sh} = G(x_r)$ (c) $|x_r - x_r^{sh}|$ (d) x_s (e) $x_s^{sh} = G(x_s)$ (f) $|x_s - x_s^{sh}|$

Figure 2.4: (a), (b) and (c) show real image x_r, translated real image x_r^{sh} and their difference $|x_r - x_r^{sh}|$ respectively. (d), (e) and (f) show synthetic image x_s, translated synthetic image x_s^{sh} and their difference $|x_s - x_s^{sh}|$ respectively.

To understand how our generative network G works, we show some examples of synthetic and real images, their transformed versions, and the difference images in Figure 2.4. This shows that G mainly operates on edges. Since depth maps are mostly discontinuous at edges, they provide important cues for the geometry of the scene. On the other hand, due to the difference between the geometry and material of objects around the edges, the rendering algorithm may find it hard to render realistic edges compared with other parts of the scene. As a result, most of the domain specific information related to geometry lies in the edges, on which SharinGAN correctly focuses.

2.3.1.4 Generalization to Make3D

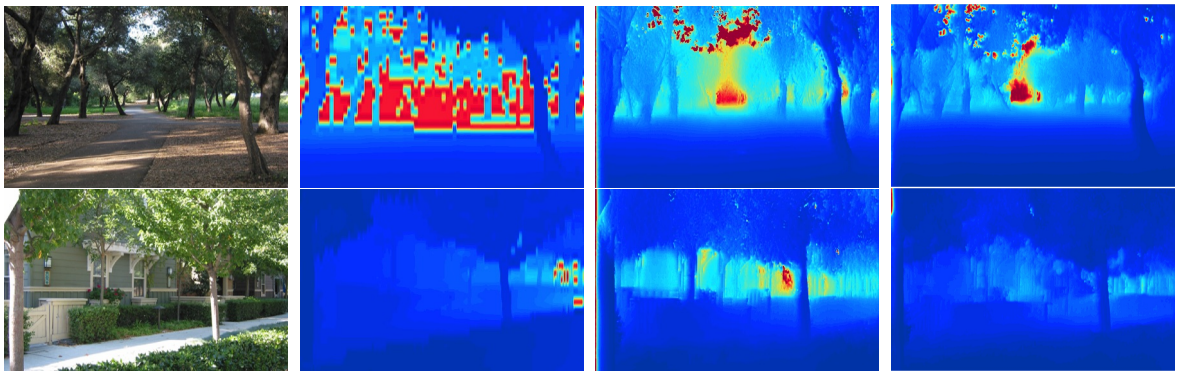
To demonstrate the generalization ability of the proposed method, we test our trained model on Make3D [31]. Note that we do not fine-tune our model using the data from Make3D. Table 2.2 shows the quantitative results of our method, which outperforms existing state-of-the-art methods by a large margin.

Moreover, the performance of SharinGAN is more comparable to the super-

Method	Trained	Error Metrics, lower is better		
		Abs Rel	Sq Rel	RMSE
Karsh et al. [65]	Yes	0.398	4.723	7.801
Laina et al. [66]	Yes	0.198	1.665	5.461
Kundu et al. [25]	Yes	0.452	5.71	9.559
Goddard et al. [67]	No	0.505	10.172	10.936
Kundu et al. [25]	No	0.647	12.341	11.567
Atapour et al. [28]	No	0.423	9.343	9.002
T2Net [26]	No	0.508	6.589	8.935
GASDA [27]	No	0.403	6.709	10.424
SharinGAN (proposed)	No	**0.377**	**4.900**	**8.388**

Table 2.2: MDE results on Make3D dataset [31]. Trained indicates whether the model is trained on Make3D or not. Errors are computed for depths less than 70m in a central image crop [67]. It can be concluded that our proposed method generalized better to an unseen dataset.

vised methods. We further visually compare the proposed method with GASDA [27] in Figure 2.5. It is clear that the proposed depth map captures more details in the input images, reflecting more accurate depth prediction.



(a) Input Image (b) Ground Truth (c) GASDA [27] (d) SharinGAN

Figure 2.5: Qualitative results on the test set of the Make3D dataset [31]. In the top row, some far tree structures that are missing in the depth map predicted by GASDA were better captured on using the SharinGAN module. For the bottom row, GASDA wrongly predicts the depth map of the houses behind the trees to be far, which is correctly captured by the SharinGAN.

2.3.2 Face Normal Estimation

2.3.2.1 Datasets

We use the synthetic data provided by [22] and CelebA [68] as real data to train the SharinGAN for face normal estimation similar to [22]. Our trained model is then evaluated on the Photoface dataset [32].

2.3.2.2 Implementation details

We use the RBDN network [69] as our generator and SfSNet [22] as the primary task network. Similar to before, we pre-train the Generator on both

Algorithm	MAE	$< 20°$	$< 25°$	$< 30°$
3DMM [52]	26.3°	4.3%	56.1%	**89.4%**
Pix2Vertex [70]	33.9°	24.8%	36.1%	47.6%
SfSNet [22]	25.5°	43.6%	57.7%	68.7%
SharinGAN (proposed)	**24.0°**	**47.88%**	**61.53%**	72.1%

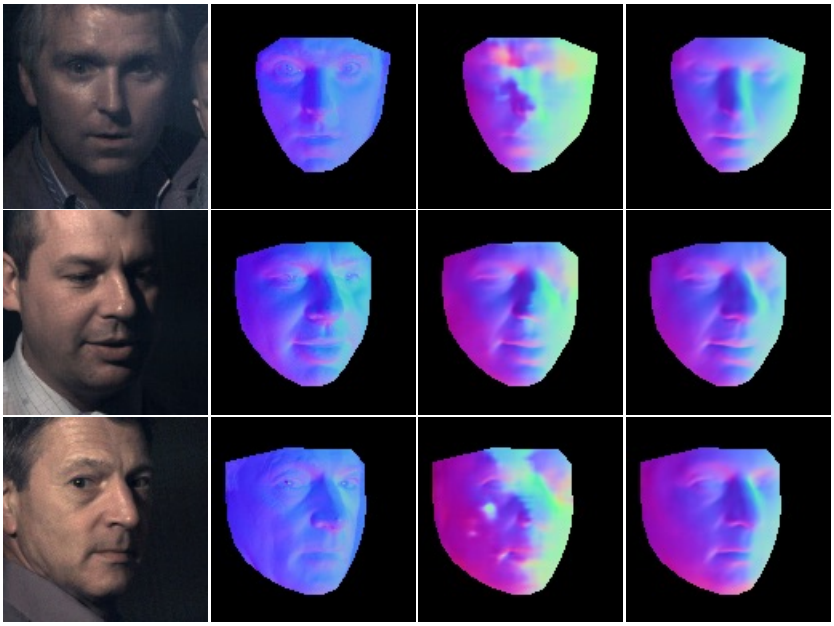
Table 2.3: Quantitative results for Face Normal estimation on the test split of Photoface dataset [32]. All the listed methods are not fine-tuned on Photoface. The metrics MAE: Mean Angular Error and $< 20°, 25°, 30°$ refer to the normals prediction accuracy for different thresholds.

synthetic and real data using reconstruction loss and pre-train the primary task network on just synthetic data in a supervised manner. Then, we train G and T end-to-end using the overall loss (2.10) for 120,000 iterations. We use a batch size of 16 and a learning rate of $1e-4$. The best model is selected based on the validation set of Photoface [32].

2.3.2.3 Results

Table 2.3 shows the quantitative performance of the estimated surface normals by our method on the test split of the Photoface dataset. With the proposed SharinGAN module, we were able to significantly improve over SfSNet on all the metrics. In particular, we were able to significantly reduce the mean angular error metric by roughly 1.5°.

Additionally, Figure 2.6 depicts the qualitative comparison of our method

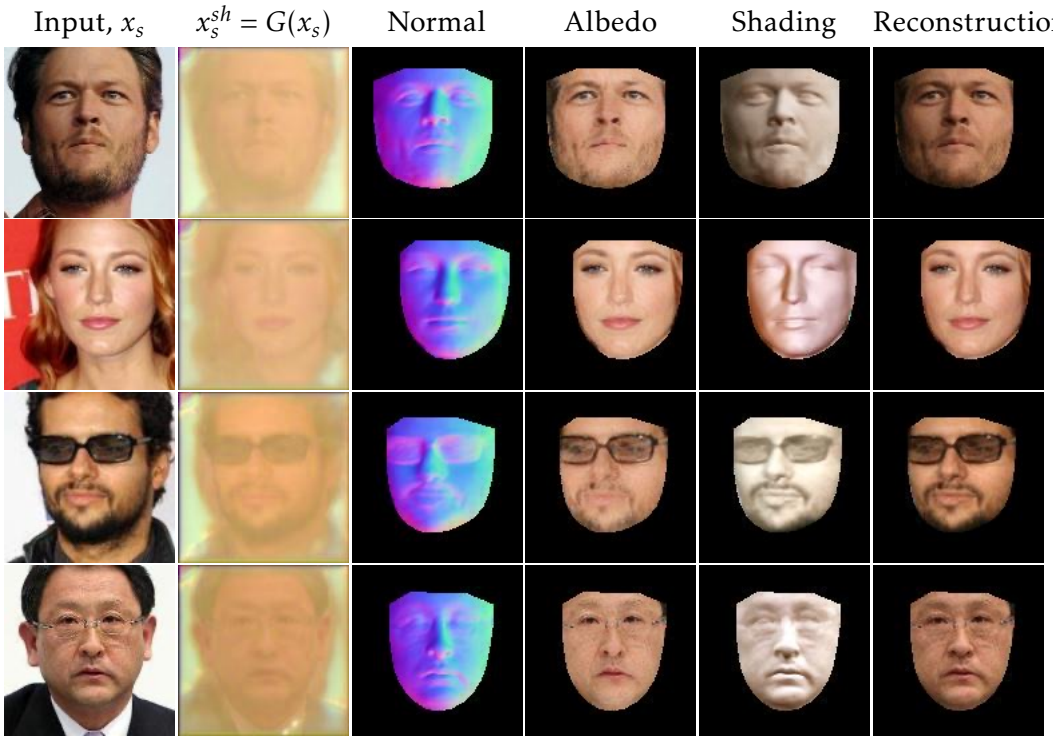


(a) Input Image (b) GT (c) SfSNet [22] (d) SharinGAN

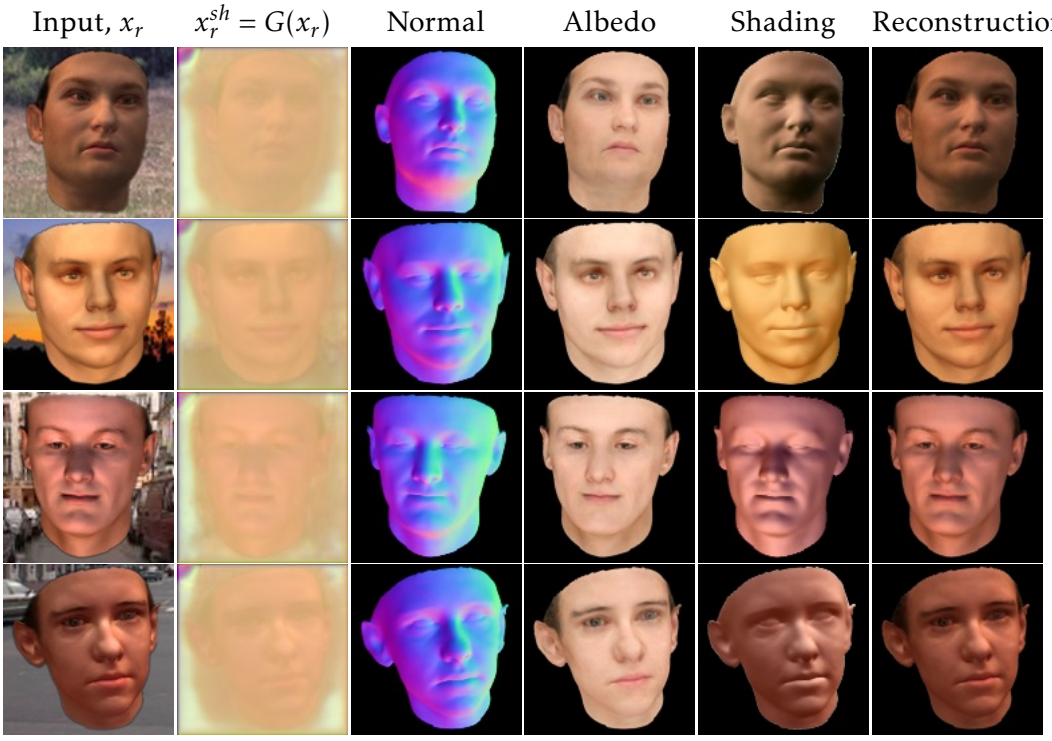
Figure 2.6: Qualitative comparisons of our method with SfSNet on the examples from the test set of Photoface dataset [32]. Our method generalizes much better to unseen data during training.

with SfSNet on the test split of Photoface. Both SfSNet and our pipeline are not finetuned on this dataset, and yet we were able to generalize better compared to SfSNet. This demonstrates the generalization capacity of the proposed SharinGAN to unseen data in training.

Finally, Figure 2.7 depicts the qualitative results of our method on the CelebA [68] and Synthetic [22] datasets. The translated images corresponding to synthetic and real images look similar in contrast to the MDE task (Figure 2.4). We suppose that for the task of MDE, regions such as edges are domain specific, and yet hold primary task related information such as depth cues, which is why



(a) Qualitative results of our method on CelebA testset [68].



(b) Qualitative results of our method on the synthetic data used in SfSNet [22].

Figure 2.7: Qualitative results of our method on face normal estimation task. The translated images x_r^{sh}, x_s^{sh} look reasonably similar for our task which additionally predicts albedo, lighting, shading and Reconstructed image along with the face normal.

SharinGAN modifies such regions. However, for the task of FNE, we additionally predict albedo, lighting, shading and a reconstructed image along with estimating normals. This means that the primary network needs a lot of shared information across domains for good generalization to real data. Thus the SharinGAN module seems to bring everything into a shared space, making the translated images $\{x_r^{sh}, x_s^{sh}\}$ look visually similar.

Lighting Estimation The primary network estimates not only face normals but also lighting. We also evaluate this. Following a similar evaluation protocol as that of [22], Table 2.4 summarizes the light classification accuracy on the MultiPIE dataset [71]. Since we do not have the exact cropped dataset that [22] used, we used our own cropping and resizing on the original MultiPIE data: centercrop 300x300 and resize to 128x128. For a fair comparison, we used the same dataset to re-evaluate the lighting performance for [22] and reported the results in Table 2.4. Our method not only outperforms [22] on the face normal estimation, but also on lighting estimation.

Algorithm	top-1%	top-2%	top-3%
SfSNet [22]	80.25	92.99	96.55
SharinGAN	**81.83**	**93.88**	**96.69**

Table 2.4: Light classification accuracy on MultiPIE dataset [71]. Training with the proposed SharinGAN also improves lighting estimation along with face normals.

2.3.3 Ablation studies

We carried out our ablation study using the KITTI and Make3D datasets on monocular depth estimation. We study the role of the SharinGAN module by removing it and training a primary network on the original synthetic and real data using (2.8). We observe that the performance drops significantly as shown in Table 2.5 and Table 2.6. This shows the importance of the SharinGAN module that helps train the primary task network efficiently.

To demonstrate the role of reconstruction loss, we remove it and train our whole pipeline $\alpha_1 L_{adv} + \alpha_3 L_T$. We show the results on the testset of KITTI in the second row of Table 2.5 and on the testset of Make3D in the second row of Table 2.6. For both the testsets, we can see the performance drop compared to our full model. Although the drop is smaller in the case of KITTI, it can be seen that the drop is significant for Make3D dataset that is unseen during training. This signifies the importance of reconstruction loss to generalize well to a domain not seen during training.

Components		Cap	Error Metrics, lower is better				Accuracy Metrics, higher is better		
SharinGAN	Reconstruction loss		Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
x	x	50m	0.137	0.804	4.12	0.210	0.816	0.940	0.978
✓	x	50m	0.1113	**0.6705**	3.80	0.192	0.861	0.954	0.980
✓	✓	50m	**0.109**	0.673	**3.77**	**0.190**	**0.864**	**0.954**	**0.981**

Table 2.5: Ablation study for monocular depth estimation to understand the role of the SharinGAN module and Reconstruction loss. We need both to get the best performance for this task.

Components		Cap	Error Metrics, lower is better		
SharinGAN	Reconstruction loss		Abs Rel	Sq Rel	RMSE
x	x	70m	0.476	8.058	9.449
✓	x	70m	0.401	5.318	**8.377**
✓	✓	70m	**0.377**	**4.900**	8.388

Table 2.6: Ablation study for monocular depth estimation to understand the role of the SharinGAN module and Reconstruction loss on the Make3D test dataset. We need both to get the best performance for this task.

2.4 Summary

Our primary motivation is to simplify the process of combining synthetic and real images in training. Prior approaches often pick one domain and try to map images into it from the other domain. Instead, we train a generator to map all images into a new, shared domain. In doing this, we note that in the new domain, the images need not be indistinguishable to the human eye, only to the network that performs the primary task. The primary network will learn to ignore extraneous, domain-specific information that is retained in the shared domain.

To achieve this, we propose a simple network architecture that rests on our new SharinGAN, which maps both real and synthetic images to a shared domain. The resulting images retain domain-specific details that do not prevent the primary network from effectively combining training data from both domains. We

demonstrate this by achieving significant improvements over state-of-the-art approaches in two important applications, surface normal estimation for faces, and monocular depth estimation for outdoor scenes. Finally, our ablation studies demonstrate the significance of the proposed SharinGAN in effectively combining synthetic and real data.

Chapter 3: LDMs for Text-Based Image Segmentation

[1]Teaching neural networks to accurately find the boundaries of objects is hard and annotation of boundaries at internet scale is impractical. Also, most self-supervised or weakly supervised problems do not incentivize learning boundaries. For example, training on classification or captioning allows models to learn the most discriminative parts of the image without focusing on boundaries [73,74]. Our insight is that Latent Diffusion Models (LDMs) [18], which can be trained without object level supervision at internet scale, must attend to object boundaries, and so we hypothesize that they can learn features which would be useful for open world image segmentation. We support this hypothesis by showing that LDMs can improve performance on this task by up to 6%, compared to standard baselines and these gains are further amplified when LDM based segmentation models are applied on AI generated images.

To test the aforementioned hypothesis about the presence of object-level semantic information inside a pretrained LDM, we conduct a simple experiment. We compute the pixel-wise norm between the unconditional and text-conditional

[1]Work done with Bharat Singh, Pallabi Ghosh, Behjat Siddiquie, and David Jacobs. Accepted [72] as ORAL in ICCV 2023.

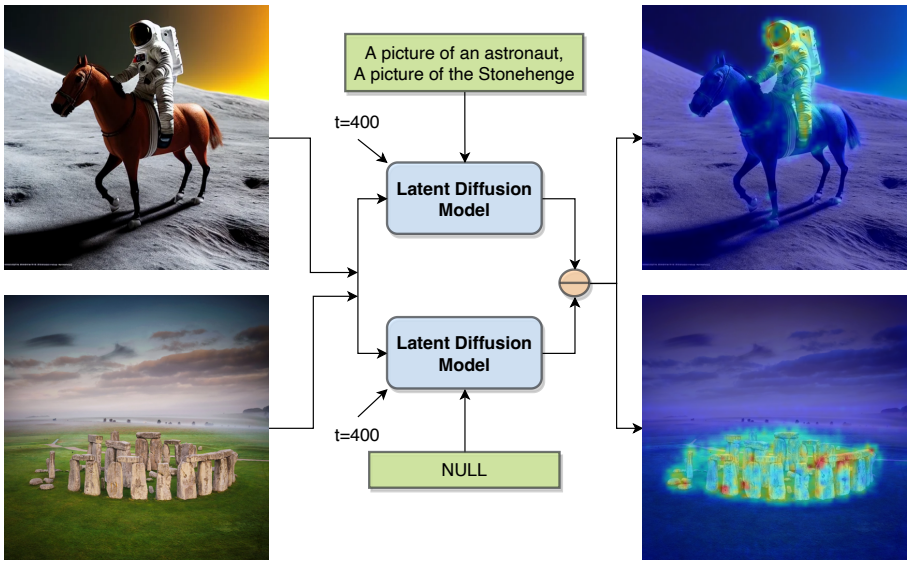


Figure 3.1: Coarse segmentation results from an LDM for two distinct images, demonstrating the encoding of fine-grained object-level semantic information within the model's internal features.

noise estimates from a pretrained LDM as part of the reverse diffusion process. This computation identifies the spatial locations that need to be modified for the noised input to align better with the corresponding text condition. Hence, the magnitude of the pixel-wise norm depicts regions that identify the text prompt. As shown in the Figure 3.1, the pixel-wise norm represents a coarse segmentation of the subject although the LDM is not trained on this task. This clearly demonstrates that these large scale LDMs can not only generate visually pleasing images, but their internal representations encode fine-grained semantic information, that can be useful for tasks like segmentation.

Recently, text-based image segmentation has gained traction for creating and editing AI generated content (like AI art, illustrations, cartoons etc.) in im-

age inpainting workflows [2] as it provides a conversational interface. Since the latent space z [75], extracted by a VQGAN is trained on several domains like art, cartoons, illustrations and real photographs, we posit that it is a more robust input representation for text-based segmentation on AI-generated images. Furthermore, the internal layers of the LDM are responsible for generating the structure of the image and hence contain rich semantic information about objects. Soft masks from these layers have also been used as a latent input in recent work on image editing [76, 77]. Since this information is already present while generating the image, we propose an architecture in the form of LD-ZNet (shown in Figure 3.3) to decode it for obtaining the semantic boundaries of objects generated in the scene. Not only does our architecture benefit segmentation of objects in AI generated images, but it also improves performance over natural images. Overall our contributions are as follows:

- We propose a text-based segmentation architecture, ZNet that operates on the compressed latent space of the LDM (z).
- Next, we study the internal representations at different stages of pretrained LDMs and show that they are useful for text-based image segmentation.
- Finally, we propose a novel approach named LD-ZNet to incorporate the visual-linguistic latent diffusion features from a pretrained LDM and show improvements across several metrics and domains for text-based image segmentation.

[2] imaginAIry, stable-diffusion-webui

3.1 Related work

3.1.1 Text-based image segmentation

Text-based image segmentation is the general task of segmenting specific regions in an image, based on a text prompt. This is different from the referring expression segmentation (RES) task, which aims to extract instance-level segmentation of different objects through distinctive referring expressions. While RES helps applications in robotics that require localization of a *single* object in an image, text-based segmentation benefits image editing applications by being able to also segment 1) "stuff" categories (clouds/ocean/beach) and 2) multiple instances of an object category applicable to the text prompt. However, both these tasks have some shared literature in terms of approaches. Preliminary works [78–82] focused on the multi-modal feature fusion between the language and visual representations obtained from recurrent networks (such as LSTM) and CNNs respectively. The subsequent set of works [83–86] included variations of multi-modal training, attention and cross-attention networks etc. Recently, [85, 87] used CLIP [88] to extract visual linguistic features of the image and the reference text separately. These features were then combined using a transformer based decoder to predict a binary mask. Alternately, [89, 90], proposed vision-language pretraining on other text-based visual recognition tasks (object detection and phrase grounding) and later finetuned for the segmentation task. The concurrent works segment-anything (SAM) [91] and segment-

everything-everywhere-all-at-once (SEEM) [92] allow interactive segmentation via point clicks, bounding boxes and text inputs . demonstrating good zero-shot performance. Different from all these works, we show the significance of using the latent space and the internal features from a pretrained latent diffusion model [18] for improving the more generic text-based image segmentation task.

3.1.2 Text-to-Image synthesis

Text-to-Image synthesis has initially been explored using GANs [39, 93–97] on publicly available image captioning datasets. Another line of work is by using autoregressive models [98–100] via a two stage approach. The first stage is a vector quantized autoencoder such as a VQVAE [101, 102] or a VQGAN [75] with an image reconstruction objective to convert an image into a shorter sequence of discrete tokens. This low dimensional latent space enables the training of compute intensive autoregressive models even for high resolution text-to-image synthesis. With the recent advancements in Diffusion Models (DM) [16, 17], both in unconditional and class conditional settings, they have started gaining more traction compared to GANs. Their success in the text-to-image tasks [103, 104] made them even more popular. However, the prior diffusion models worked in the high-dimensional image space that made training and inference computationally intensive. Subsequently, latent space representations [18, 105–107] were proposed for high resolution text-to-image synthesis to reduce the heavy compute demands. More specifically, the latent diffusion model (LDM) [18]

mitigates this problem by relying on a perceptually compressed latent space produced by a powerful autoencoder from the first stage. Moreover, they employ a convolutional backed UNet [108] as the denoising architecture, allowing for different sized latent spaces as input. Recently this architecture is trained on large scale text-image data [109] from the internet and released as Stable-diffusion[3], which exhibited photo-realistic image generations. Subsequently, several language guided image editing applications such as inpainting [110–112], text-guided image editing [77, 113] became more popular and the usage for text-based image segmentation has surged, especially for AI generated images. We propose a solution for text-based image segmentation by leveraging the features which are already present as part of the synthesis process.

3.1.3 Semantics in generative models

Semantics in generative models such as GANs have been studied for binary segmentation [114, 115] as well as multi-class segmentation [3, 4, 116] where the intermediate features have been shown to contain semantic information for these tasks. Moreover, [117] highlighted the practical advantages of these representations, such as out-of-distribution robustness. However, prior generative models (GANs) as representation learners have received less attention compared to alternative unsupervised methods [118], because of the training difficulties on complex, diverse and large scale datasets. Diffusion models [16], on the other hand are another class of powerful generative models that recently outperformed

[3] https://github.com/CompVis/stable-diffusion

GANs on image synthesis [17] and are able to train on large datasets such as Imagenet [119] or LAION [109]. In [5], the authors demonstrated that the internal features of a pre-trained diffusion model were effective at the semantic segmentation task. However, this type of analysis [4,5] has mostly been done in limited settings like few shot learning [120] or limited domains like faces [121], horses [122] or cars [122]. Different from these works, we analyze the visual-linguistic semantic information present in the internal features of a text-to-image LDM [18] for text based image segmentation, which is an open world visual recognition task. Furthermore, we leverage these LDM features and show performance improvements when training with full datasets instead of few-shot settings.

3.2 LDMs for Text-Based Segmentation

The text-to-image latent diffusion architecture introduced in [18] consists of two stages: 1) An auto-encoder based VQGAN [75] that extracts a compressed latent representation (z) for a given image 2) A diffusion UNet that is trained to denoise the noisy z created in the forward diffusion process, conditioned on the text features. These text features are obtained from a pretrained frozen CLIP text encoder [88] and is conditioned at multiple layers of the UNet via cross-attention.

In this paper, we show performance improvements on the text-based segmentation task in two steps. Firstly, we analyze the compressed latent space (z) from the first-stage and propose an approach named ZNet that uses z as the visual input to estimate segmentation mask when conditioned on a text prompt. Sec-

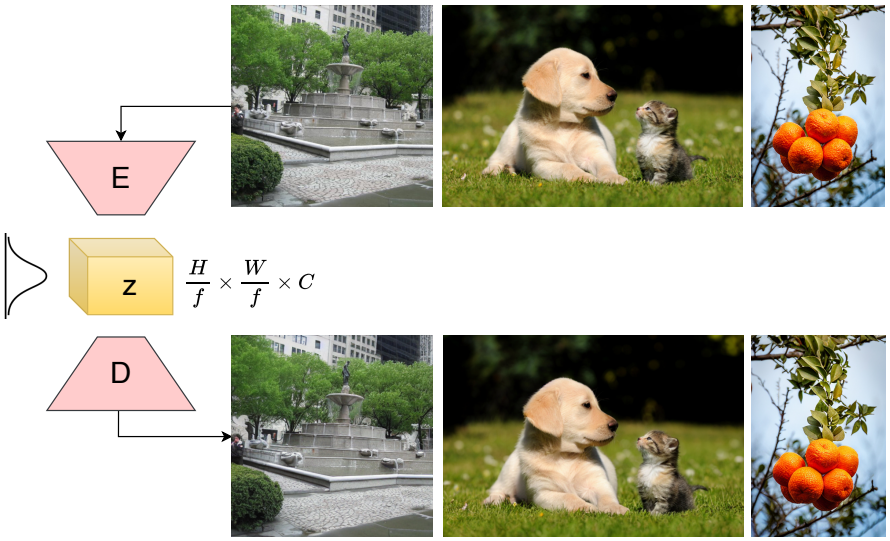


Figure 3.2: Reconstructions from the first stage of the LDM. Given an input image, the latent representation z generated by the encoder, can be used to reconstruct images that are perceptually indistinguishable from the inputs. The high quality of these reconstructions suggests that the latent representation z, preserves most of the semantic information present in the input images.

ondly, we study the internal representations from the second stage of the stable-diffusion LDM for visual-linguistic semantic information and propose a way to utilize them inside ZNet for further improvements in the segmentation task. We name this approach as LD-ZNet.

3.2.1 ZNet: Leveraging Latent Space Features

We observe that the latent space (z) from the first-stage of the LDM is a compressed representation of the image that preserves semantic information, as depicted in Figure 3.2. The VQGAN in the first-stage achieves such semantic-preserving compression with the help of large scale training data as well as a

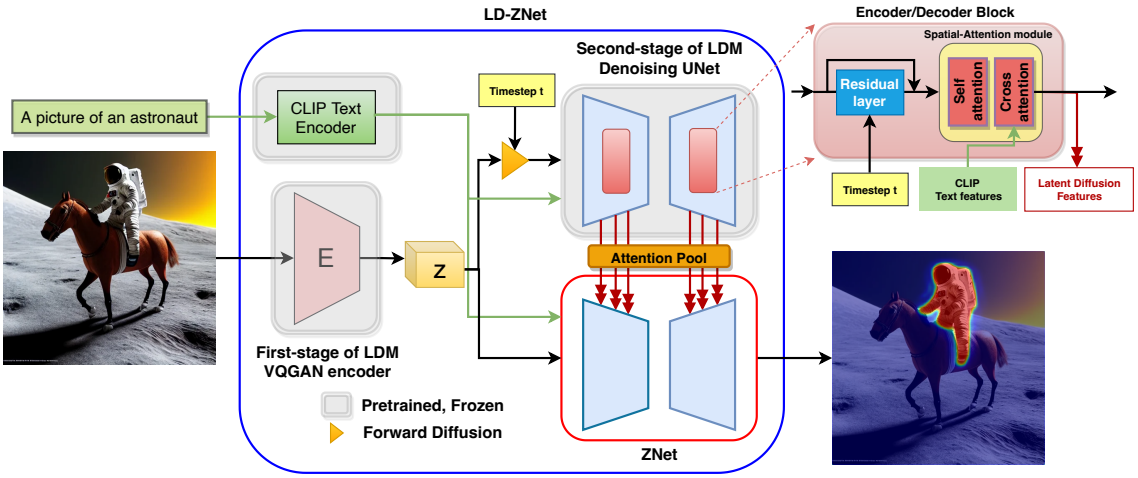


Figure 3.3: Overview of the proposed ZNet and LD-ZNet architectures. We propose to use the compressed latent representation z as input for our segmentation network ZNet. Next, we propose LD-ZNet, which incorporates the latent diffusion features at various intermediate blocks from the LDM's denoising UNet, into ZNet.

combination of losses - perceptual loss [123], a patch-based [124] adversarial objective [75, 125, 126], and a KL-regularization loss.

In our experiments, we observe that this compressed latent representation z is more robust compared to the original image in terms of their association with the text prompts. We believe this is because z is a $\frac{H}{8} \times \frac{W}{8} \times 4$ dimensional feature with 48 $\times$ fewer elements compared to the original image, while preserving the semantic information. Several prior works [127–129], show that compression techniques like PCA, which create information preserving lower dimensional representations generalize better. Therefore, we propose using the z representation along with the frozen CLIP text features [88] as an input to our segmentation network. Furthermore, because the VQGAN is trained across sev-

eral domains like art, cartoons, illustrations, portraits, etc., it learns a robust and compact representation which generalizes better across domains, as can be seen in our experiments on AI generated images. We call this approach ZNet. The architecture of ZNet is shown in the bottom box of Figure 3.3, and is the same as the denoising UNet module of the LDM. We therefore initialize it with pretrained weights of the second-stage of the LDM.

3.2.2 LD-ZNet: Leveraging Diffusion Features

Given a text prompt and a timestep t, the second-stage of the LDM is trained to denoise z_t - a noisy version of the latent representation z obtained via forward diffusion process for t timesteps. A UNet architecture is used whose encoder/decoder elements are shown in Figure 3.3 (top right). A typical encoder/decoder block contains a residual layer followed by a spatial-attention module that internally has self-attention and then cross-attention with the text features. We analyze the semantic information in the internal visual-linguistic representations developed at different blocks of encoder and decoder right after these spatial-attention modules. We also propose a way to utilize these latent diffusion features using cross-attention into the ZNet segmentation network and we call the final model as LD-ZNet.

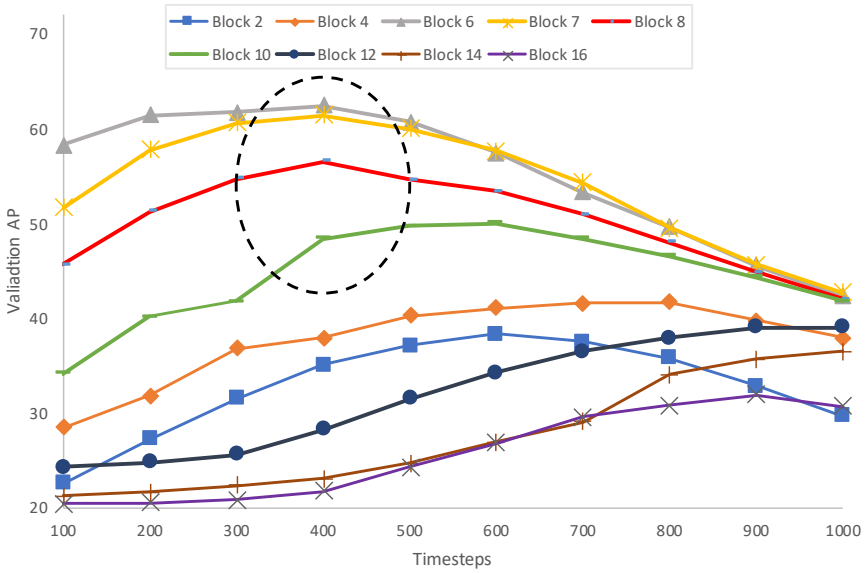


Figure 3.4: Semantic information present in the LDM features at various blocks and timesteps for the referring image segmentation task. AP is measured on a small validation subset of the PhraseCut dataset.

3.2.2.1 Visual-Linguistic Information in LDM Features

We evaluate the semantic information present in the pretrained LDM at various blocks and timesteps for the text-based image segmentation task. In this experiment, we consider the latent diffusion features right after the spatial-attention layers 1-16 spanning across all the encoder and decoder blocks present in the UNet. At each block, we analyze the features for every 100^{th} timestep in the range $[100, 1000]$. We use a small subset of the training and validation sets from the Phrasecut dataset and train a simple decoder on top of these features to predict the associated binary mask. Specifically, given an image I and timestep t, we first extract its latent representation z from the first stage of LDM and add noise from the forward diffusion to obtain z_t for a timestep t. Next we extract

the frozen CLIP text features for the text prompt and input both of them into the denoising UNet of the LDM to extract the internal visual-linguistic features at all the blocks for that timestep. We use these representations to train the corresponding decoders until convergence. Finally, we evaluate the AP metric on a small subset of the validation dataset. The performance of features from different blocks and timesteps is shown in Figure 3.4.

Similar to [5], we observe that the middle blocks {6,7,8,9,10} of the UNet contain more semantic information compared to either the early blocks of the encoder or the later blocks of the decoder. We also observe that the timesteps 300-500 contain the maximum visual-linguistic semantic information compared to other timesteps, for these middle blocks. This is in contrast to the findings of [5] that report the timesteps {50, 150, 250} to contain the most useful information when evaluated on an unconditional DDPM model for the few shot semantic segmentation task for horses [122] and faces [121]. We believe that the reason for this difference is because, in our case, the image synthesis is guided by text, leading to the emergence of semantic information earlier in the reverse diffusion process (t=1000→0), in contrast to unconditional image synthesis.

3.2.2.2 LD-ZNet Architecture

We propose using the aforementioned visual-linguistic representations at multiple spatial-attention modules of the pretrained LDM into the ZNet as shown in Figure 3.3. These latent diffusion features are injected into the ZNet via a

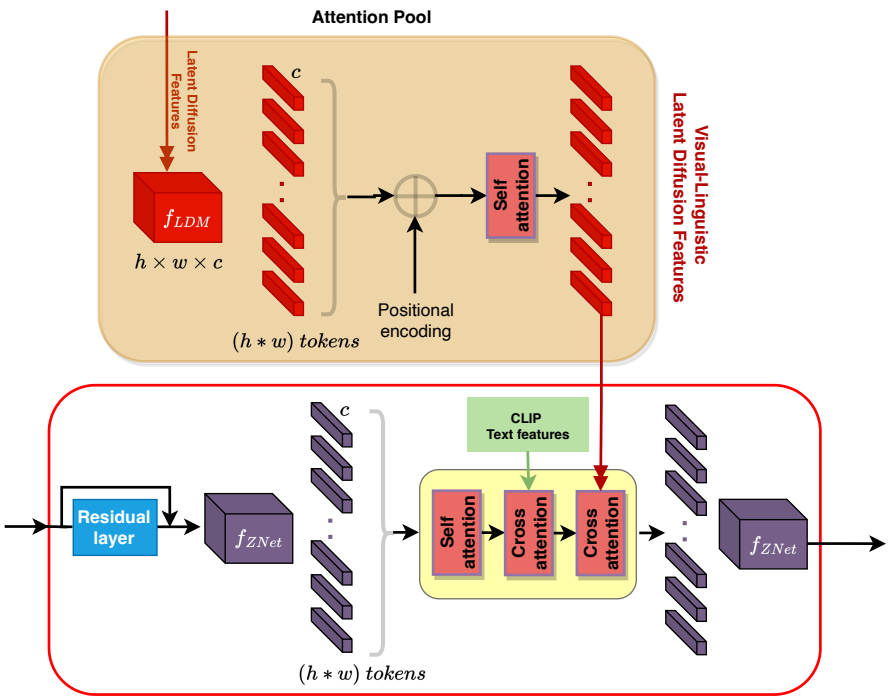


Figure 3.5: We propose to incorporate the visual-linguistic representations from LDM obtained at the spatial-attention modules via a cross-attention mechanism into the corresponding spatial-attention modules of the ZNet through an *attention pool* layer.

cross-attention mechanism at the corresponding spatial-attention modules as shown in Figure 3.5. This allows for an interaction between the visual-linguistic representations from the ZNet and the LDM. Specifically, we pass the latent diffusion features through an *attention pool* layer that not only acts as a learnable layer to match the range of the features participating in the cross-attention, but also adds a positional encoding to the pixels in the LDM representations. The outputs from the attention pool are now positional-encoded visual-linguistic representations that enable the proposed cross-attention mechanism to attend to the corresponding pixels from the ZNet features. ZNet when augmented with these

latent diffusion features from the LDM (through cross-attention) is referred to as LD-ZNet.

Following the semantic analysis of latent diffusion features (Sec. 3.2.2.1), we incorporate the internal features from blocks {6,7,8,9,10} of the LDM into the corresponding blocks of ZNet, in order to make use of the maximum semantic and diverse visual-linguistic information from the LDM. For AI generated images, these blocks are anyways responsible to generate the final image and using LD-ZNet, we are able to tap into this information which can be used for segmenting objects in the scene.

3.3 Experiments

Implementation details: In this paper, we use the stable-diffusion v1.4 checkpoint as our LDM that internally uses the frozen ViT-L/14 CLIP text encoder [88]. We implement the above described ZNet and LD-ZNet in pytorch inside the stable-diffusion library. We also initialize our networks with the weights from the LDM wherever possible, while initializing the remaining parameters from a normal distribution. We train ZNet and LD-ZNet on 8 NVIDIA A100 gpus with a batch size of 4 using the Adam optimizer and a base learning rate of $5e^{-7}$ per mini-batch sample, per gpu. For all our experiments, we keep the text encoder frozen and use an image resolution of 384 for a fair comparison with the previous works.

Datasets: We use Phrasecut [130], which is currently the largest dataset for

Figure 3.6: Samples from AIGI dataset along with annotated labels and categorical captions.

the *text-based image segmentation* task, with nearly 340K phrases along with corresponding segmentation masks that not only permit annotations for stuff classes but also accommodate multiple instances. Following [88], we randomly augment the phrases from a fixed set of prefixes. For the images, we randomly crop a square around the object of interest with maximum area, ensuring that the object remains at least partially visible. We avoid negative samples to remove ambiguity in the LDM features for non-existent objects.

We create a dataset consisting of AI-generated images which we name **AIGI** dataset, to showcase the usefulness of our approach for text-based segmentation on a different domain. We use 100 AI-generated images from *lexica.art* and man-

ually annotated multiple regions for 214 text-prompts relevant to these images. Figure 3.6 depicts some of the images from the AIGI dataset along with their annotated labels and categorical captions.

We also use the popular referring expression segmentation datasets namely RefCOCO [131], RefCOCO+ [131] and G-Ref [132] to demonstrate the generalization abilities of ZNet and LD-ZNet. In RefCOCO, each image contains two or more objects and each expression has an average length of 3.6 words. RefCOCO+ is derived from RefCOCO by excluding certain absolute-location words and focuses on purely appearance based descriptions. For example it uses "the man in the yellow polka-dotted shirt" rather than "the second man from the left" which makes it more challenging. Unlike RefCOCO and RefCOCO+, the average length of sentences in G-Ref is 8.4 words, which have more words about locations and appearances. While we adopt the UNC partition for RefCOCO and RefCOCO+ in this paper, we use the UMD partition for G-Ref.

Metrics: We follow the evaluation methodology of [87] and report best foreground IoU (IoU_{FG}) for the foreground pixels, the best mean IoU of all pixels (mIoU), and the Average Precision (AP).

3.4 Results

3.4.1 Image Segmentation Using Text Prompts

On the PhraseCut dataset, we compare the performance of previous approaches with our ZNet and LD-ZNet for the text-based image segmentation

Method	mIoU	IoU_{FG}	AP
MDETR [89]	53.7	-	-
GLIPv2-T [90]	59.4	-	-
RMI [130]	21.1	42.5	-
Mask-RCNN Top [130]	39.4	47.4	-
HulaNet [130]	41.3	50.8	-
CLIPSeg (PC+) [87]	43.4	54.7	76.7
CLIPSeg (PC, D=128) [87]	48.2	56.5	78.2
RGBNet	46.7	56.2	77.2
ZNet (Ours)	51.3	59.0	78.7
LD-ZNet (Ours)	**52.7**	**60.0**	**78.9**

Table 3.1: Text-based image segmentation performance on the PhraseCut testset. The performance of ZNet and LD-ZNet is highlighted in gray. Both these models outperform the baseline RGBNet on all the metrics.

task (Table 3.1). In order to showcase the performance improvement of our proposed networks, we create a baseline named RGBNet with the same architecture as ZNet except we use the original images as the input instead of its latent space z. For RGBNet, we use additional learnable convolutional layers to map the original image to match the input resolution of ZNet. From Table 3.1, we observe that our ZNet and LD-ZNet significantly outperform RGBNet. Specifically, the performance improvement from using the latent representation z over the origi-

nal images is clear (i.e. ZNet vs RGBNet baseline). Performance further improves upon incorporating the LDM visual-linguistic representations (LD-ZNet) - by 6% overall on the $mIoU$ metric compared to RGBNet. We also highlight this qualitatively in Figure 3.7. In the figure, we show the original image and the GT mask along with outputs from the RGBNet baseline followed by ZNet and LD-ZNet, where both ZNet and LD-ZNet help improve results consistently. For example in the top row, RGBNet detects light fixtures for the "hanging clock" prompt, and although ZNet does not have as strong activations for these incorrect detections, it is LD-ZNet that correctly segments the "clock". Similarly in the bottom row, while RGBNet completely got the "castle" wrong, ZNet correctly has activations on the right buildings, but with lower confidence. However, LD-ZNet improves it further.

We outperform in all the metrics when compared to previous works, other than MDETR [89] and GLIPv2 [90]. Notably, these works are pre-trained on detection and phrase grounding for predicting bounding boxes on huge corpus of text-image pairs across various publicly available datasets with bounding box annotations and are later fine-tuned on the Phrasecut dataset for the segmentation task. However, our work is orthogonally focused towards exploring and utilizing LDMs and its internal features for improving the text-based segmentation performance. Note that object detection datasets have a good overlap with the visual content in PhraseCut, however, they are not representative of the diversity in images available on the internet. For example, while they could learn common concepts like sky, ocean, chair, table and their synonyms, methods like MDETR

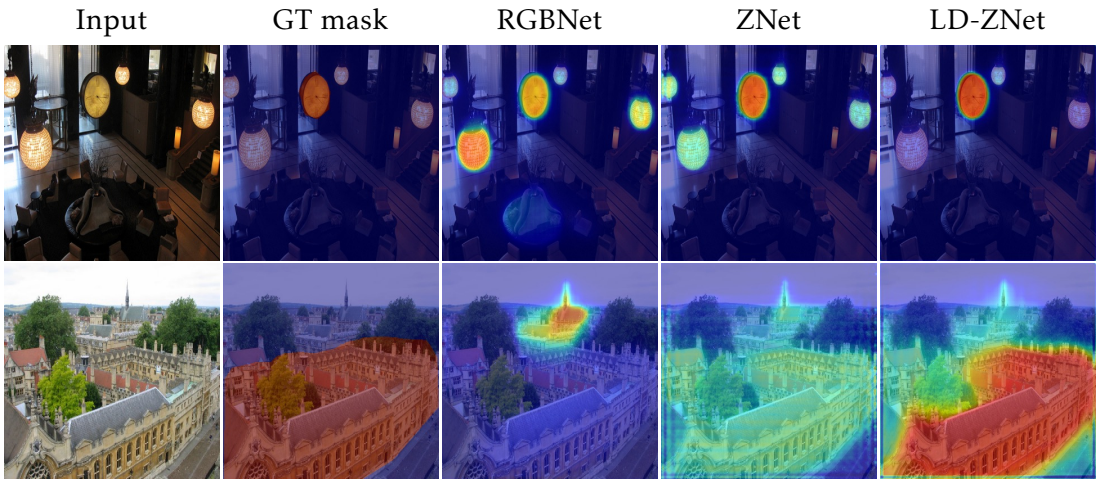


Figure 3.7: Qualitative comparison on the PhraseCut test set. Each row contains an input image with a text prompt as an input, with the goal being to segment the image regions corresponding to the reference text. The text prompts are *"hanging clock"* and *"castle"* for the top and bottom rows. We show improvements using ZNet and LD-ZNet compared to the RGBNet.

would not understand concepts like Mikey Mouse, Pikachu etc., which we will show in Section 3.5.

3.4.2 Generalization to AI Generated Images

With the growing popularity of AI generated images, text-based image segmentation is extensively being used by content creators in their daily workflows. Many public libraries [4] widely employ methods such as CLIPSeg [87] for performing segmentation in AI-generated images. So we study the generalization ability of our proposed segmentation approach on AI-generated images. To this

[4]imaginAIry, stable-diffusion-webui

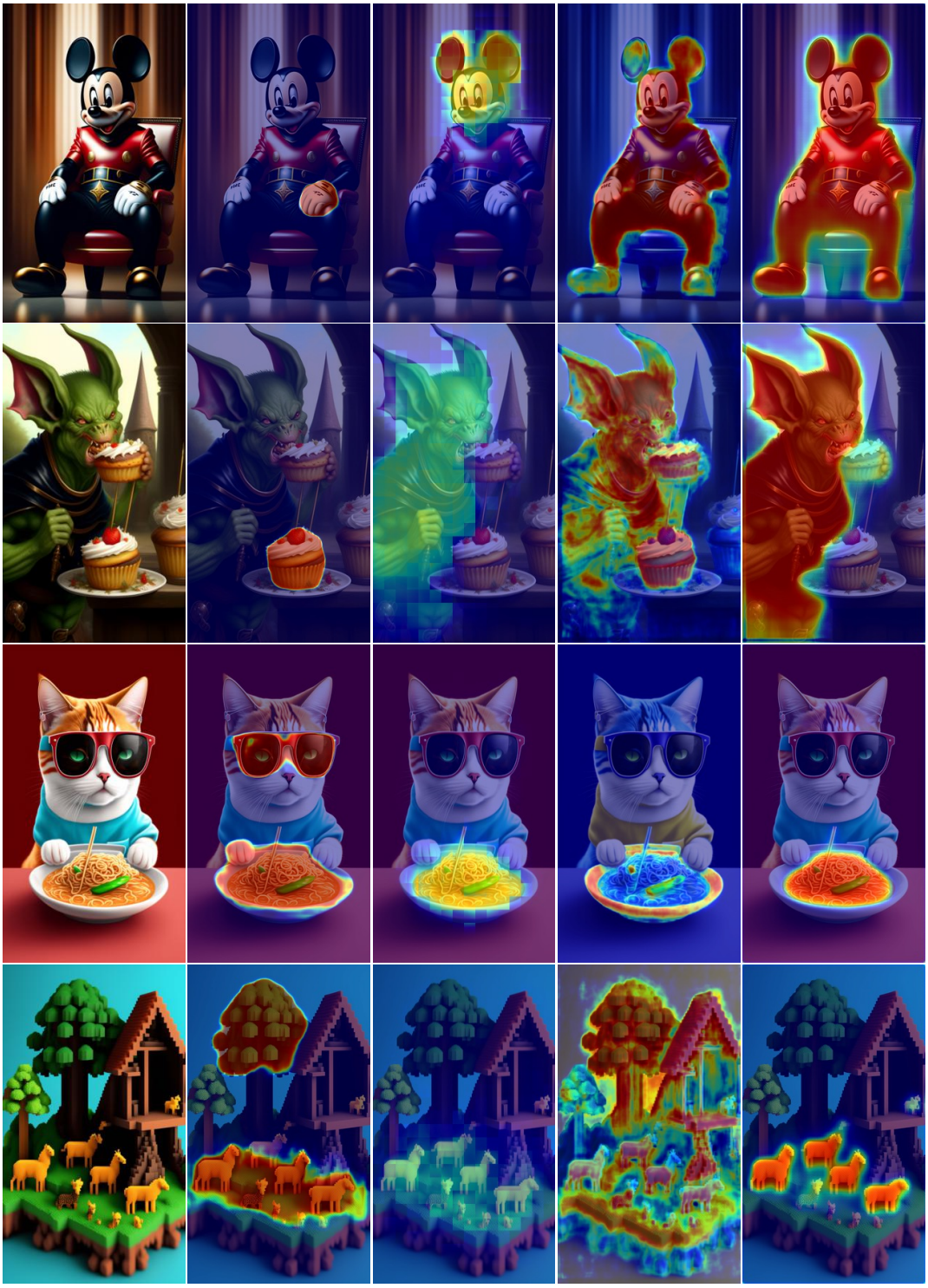
Method	mIoU	AP
MDETR [89]	53.4	63.8
CLIPSeg (PC+) [87]	56.4	79.0
SEEM [92]	57.4	70.0
RGBNet	63.4	84.1
ZNet (Ours)	68.4	85.0
LD-ZNet (Ours)	**74.1**	**89.6**

Table 3.2: Generalization of the proposed LD-ZNet on our AIGI dataset when compared with other state-of-the-art text-based segmentation methods.

extent, we first prepare a dataset of 100 AI-generated images from lexica.art and manually annotate them using 214 text-prompts. We name this dataset AIGI and release it on our project website [5] for future research. Next, we evaluate our approaches ZNet and LD-ZNet along with our RGBNet baseline and other text-based segmentation methods - CLIPSeg (PC+) [87], MDETR [89] and SEEM [92]. Glipv2 and the SAM model [91] with textual input were not publicly available for us to evaluate at the time of this work. All these methods are trained on the Phrasecut dataset except for SEEM and we measure the IoU metric as shown in Table 3.2. It can be seen that RGBNet outperforms CLIPSeg, MDETR and SEEM because its built on the UNet architecture initialized from the LDM weights that contains semantic information for good generalization. Our methods ZNet and LD-ZNet further improve the generalization to these AI-generated images by

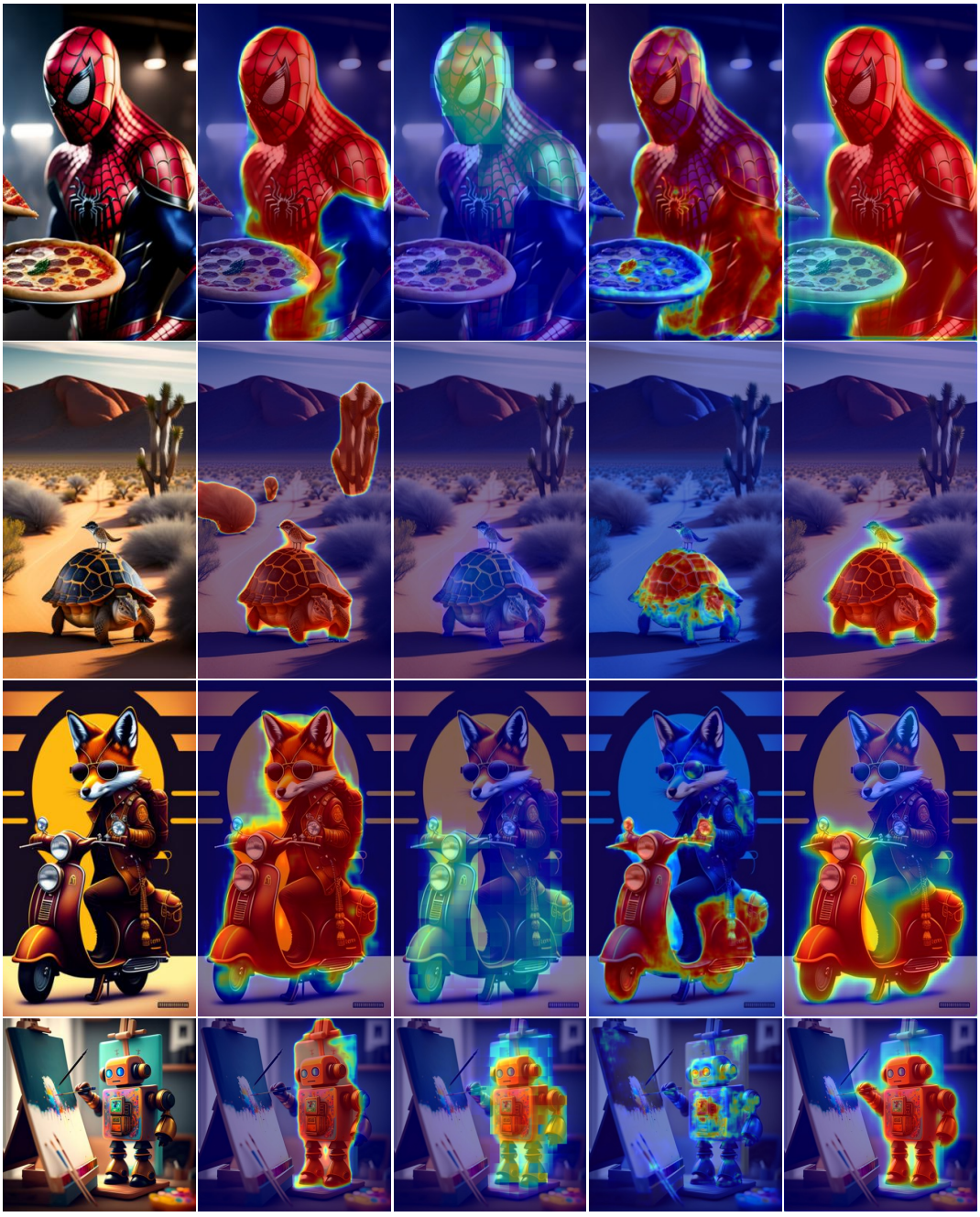
[5] https://koutilya-pnvr.github.io/LD-ZNet/

more than 20% compared to MDETR. This is largely due to the robust z-space of the LDM that resulted from a VQGAN pre-training on a variety of domains like art, cartoons, illustrations . Furthermore, the latent diffusion features that contain useful semantic information for the synthesis task, also help in segmenting the AI-generated images. We show the qualitative comparison of these methods in Figure 3.8 for four AI-generated images from our dataset. While CLIPSeg can estimate most distinctive regions such as face of the *Mickey mouse* or rough locations of *Goblin*, *Ramen* and *animals*, MDETR and SEEM incorrectly segment them because these concepts are unknown to them and because of the domain gap between their training data and AIGI images respectively. In both such cases, our proposed LD-ZNet estimates accurate segmentation. More qualitative results for LD-ZNet on images from the AIGI dataset are shown in Figs. 3.9 and 3.10.



Input MDETR [89] CLIPSeg [87] SEEM [92] LD-ZNet

Figure 3.8: Qualitative comparison on the AI-generated images for text-based segmentation. The text prompts are *"Mickey mouse"*, *"Goblin"*, *"Ramen"* and *"animals"* respectively.



Input MDETR [89] CLIPSeg [87] SEEM [92] LD-ZNet

Figure 3.9: More qualitative comparison on the AI-generated images from AIGI dataset for text-based segmentation. The text prompts are *"Spiderman"*, *"tortoise"*, *"vespa"* and *"robot"* respectively.

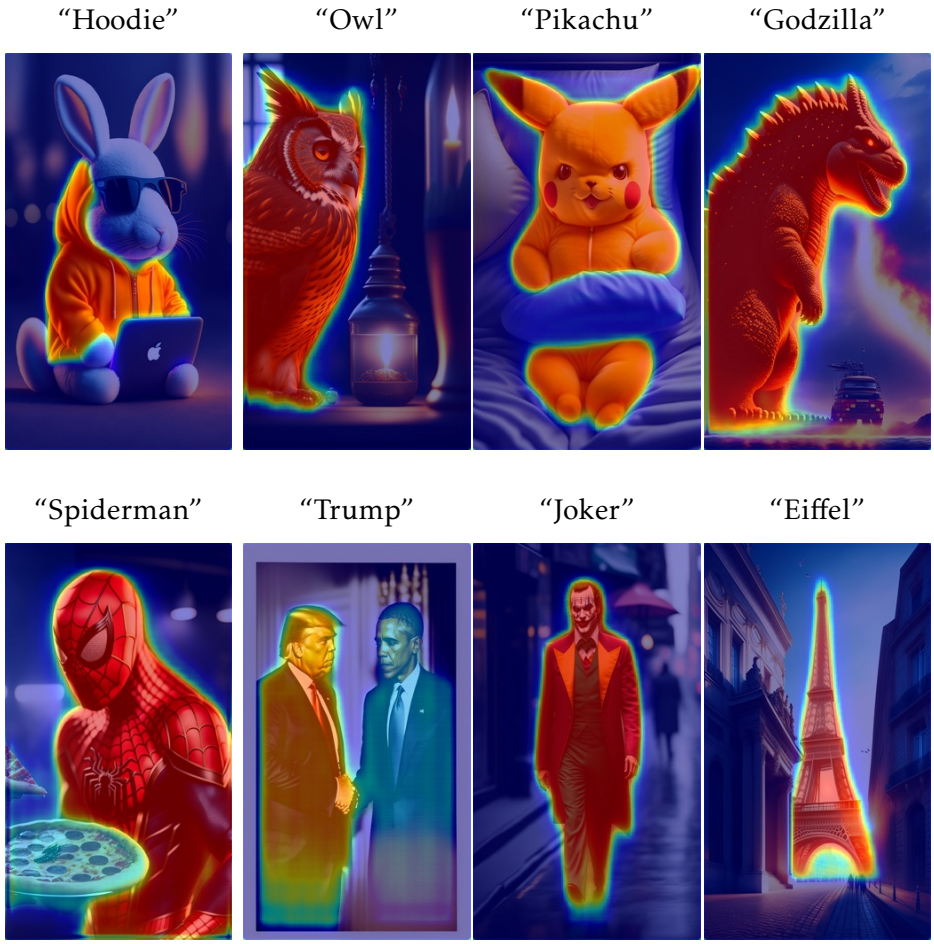


Figure 3.10: More qualitative results of LD-ZNet from AIGI dataset.

3.4.3 Generalization to Referring Expressions

The reference expression segmentation task is aimed at robot-localization types of applications, where segmenting at the instance-level is performed through distinctive referring expressions. Many works such as [85, 86] also train the text encoder to learn the complex positional references in the text. However, we are focused on generic text-based segmentation that has support for stuff categories as well as for multiple instances. We study the generalization ability of the

Method	RefCOCO		RefCOCO+		G-Ref	
	IoU	AP	IoU	AP	IoU	AP
CLIPSeg (PC+) [87]	30.1	14.1	30.3	15.5	33.8	23.7
RGBNet	36.3	15.7	37.1	16.7	41.9	27.8
ZNet (Ours)	40.1	16.8	40.9	17.8	47.1	29.2
LD-ZNet (Ours)	**41.0**	**17.2**	**42.5**	**18.6**	**47.8**	**30.8**

Table 3.3: Generalization of our proposed approaches to different types of expressions from other datasets. Z-Net and LD-ZNet outperform both the RGBNet baseline and CLIPSeg on the generalization across all datasets.

proposed approach - using LDM features, to this complex task. Specifically, we use the models trained on the PhraseCut dataset and evaluate them on the RefCOCO [131], RefCOCO+ [131] and G-Ref [132] datasets whose complex referring expressions are for single-instance localization and segmentation. We also evaluated the generalization of the CLIPSeg (PC+) [87] model that was trained on an extended version of the PhraseCut dataset (PC+), to further demonstrate the generalization capability of our methods. Table 3.3 summarizes the performance of our models along with the RGBNet baseline. We observe a similar trend in performance improvements across RGBNet < *Z*Net < LD-ZNet. These experiments demonstrate that the LDM features enhance the generalization power of the LD-ZNet model even on complex referring expressions.

3.4.4 Inference Time

During inference, our proposed LD-ZNet relies on the LDM to extract the internal features for just a single time step (as opposed to around 50 reverse diffusion time steps for the text-to-image synthesis task). We then use these LDM features for further cross-attention into LD-ZNet via the attention pool layer to extract the final mask. Therefore, using the diffusion model increases the overall run time by only a small amount. For the stable-diffusion model, inference takes 2.57s for 50 timesteps to synthesize an image (roughly 51ms per timestep), whereas the average inference times for RGBNet, ZNet and LD-ZNet are only 62ms, 55ms and 101ms, respectively, per image on the AIGI dataset with an RTX A6000 gpu. SEEM [92] takes 293ms for the same task. Since we use an architecture similar to UNet (from the second stage of the LDM), as our segmentation network, the proposed LD-ZNet has 925M trainable parameters.

3.4.5 Cross-attention vs Concat for LDM features

In LD-ZNet, we inject LDM features into the ZNet model using cross-attention (Figure 3.5). In order to understand the importance of the cross-attention layer, we also train and evaluate another model where the LDM features are concatenated with the features of the ZNet right before the spatial-attention layer. The results are summarized in Table 3.4 and it shows that concatenating the LDM features yields inferior results compared to the proposed method. This is because of the *attention pool* layer which serves as a learnable layer and also encodes

Diffusion features via	mIoU	IoU_{FG}	AP
LD-ZNet with concatenation	50.2	59.0	78.1
LD-ZNet with cross-attention	**52.7**	**60.0**	**78.9**

Table 3.4: Incorporating LDM features into ZNet via cross-attention (LD-ZNet) leverages the visual-linguistic information present in them, compared to concatenation, leading to better performance on the text-based image segmentation task.

positional information into the LDM features for setting up the cross-attention. Moreover, the cross-attention layer learns how feature pixels from the ZNet attend to feature pixels from the LDM, thereby leveraging context and correlations from the entire image. With concatenation however, we only fuse the corresponding features of LDM and ZNet which is sub-optimal.

3.5 Discussion

In this section we present more qualitative results to demonstrate several interesting aspects of our proposed technique when applied towards downstream segmentation tasks. In Figs. 3.8 to 3.12, we visualize results of text-based image segmentation on a diverse set of images, which include AI generated images, illustrations and generic photographs. In Figure 3.11, we show that when LD-ZNet is applied on the same image with various text prompts, it is able to correctly segment the object and stuff classes being referred to in both examples. This capability is crucial for open-world segmentation and overall under-

standing of the scene. The results also highlights that the algorithm works remarkably well on other domains like cartoons/illustrations. It is noteworthy that LD-ZNet can perform accurate segmentation for text prompts which include cartoons (Pikachu, Godzilla), celebrities (Donald Trump, Spiderman), famous landmarks (Eiffel Tower), as seen in Figure 3.10. Finally, Figure 3.12 shows the advantages of leveraging semantic information present in the latent diffusion features. Compared to our baseline RGBNet, the proposed LD-ZNet generates better segmentation maps across animations, celebrity images and illustrations.

3.6 Summary

In this chapter, we presented a novel approach for text-based image segmentation using large scale latent diffusion models. By training the segmentation models on the latent z-space, we were able to improve the generalization of segmentation models to new domains, like AI generated images. We also showed that this z-space is a better representation for text-to-image tasks in natural images. By utilizing the internal features of the LDM at appropriate time-steps, we were able to tap into the semantic information hidden inside the image synthesis pipeline using a cross-attention mechanism, which further improved the segmentation performance both on natural and AI generated images. This was experimentally validated on several publicly available datasets and on a new dataset of AI generated images, which we will make publicly available.

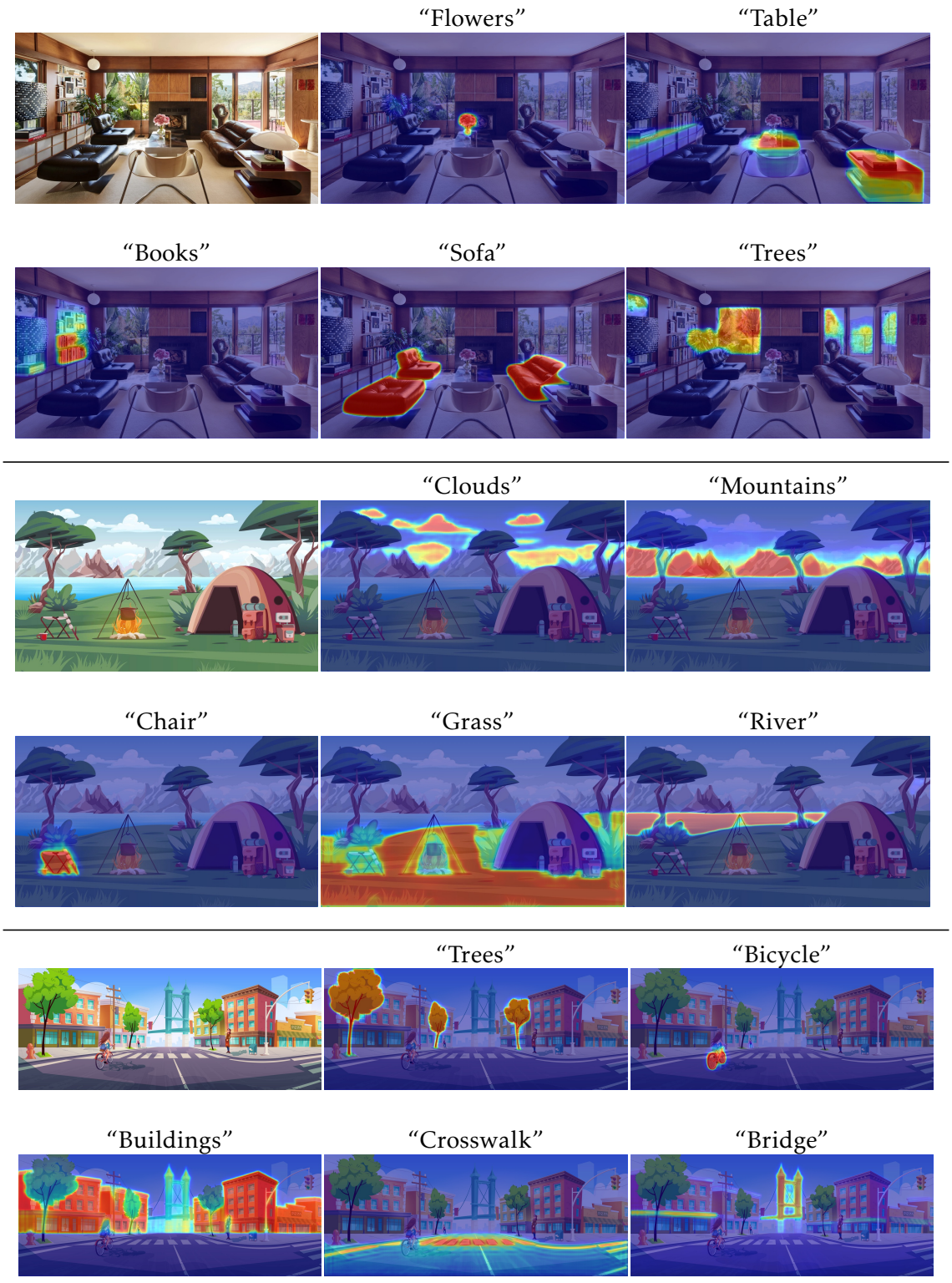


Figure 3.11: LD-ZNet text-based image segmentation results for a real image and illustrations on diverse set of things and stuff classes. High quality segmentation across multiple classes suggests that LD-ZNet has a good understanding of the overall scene.

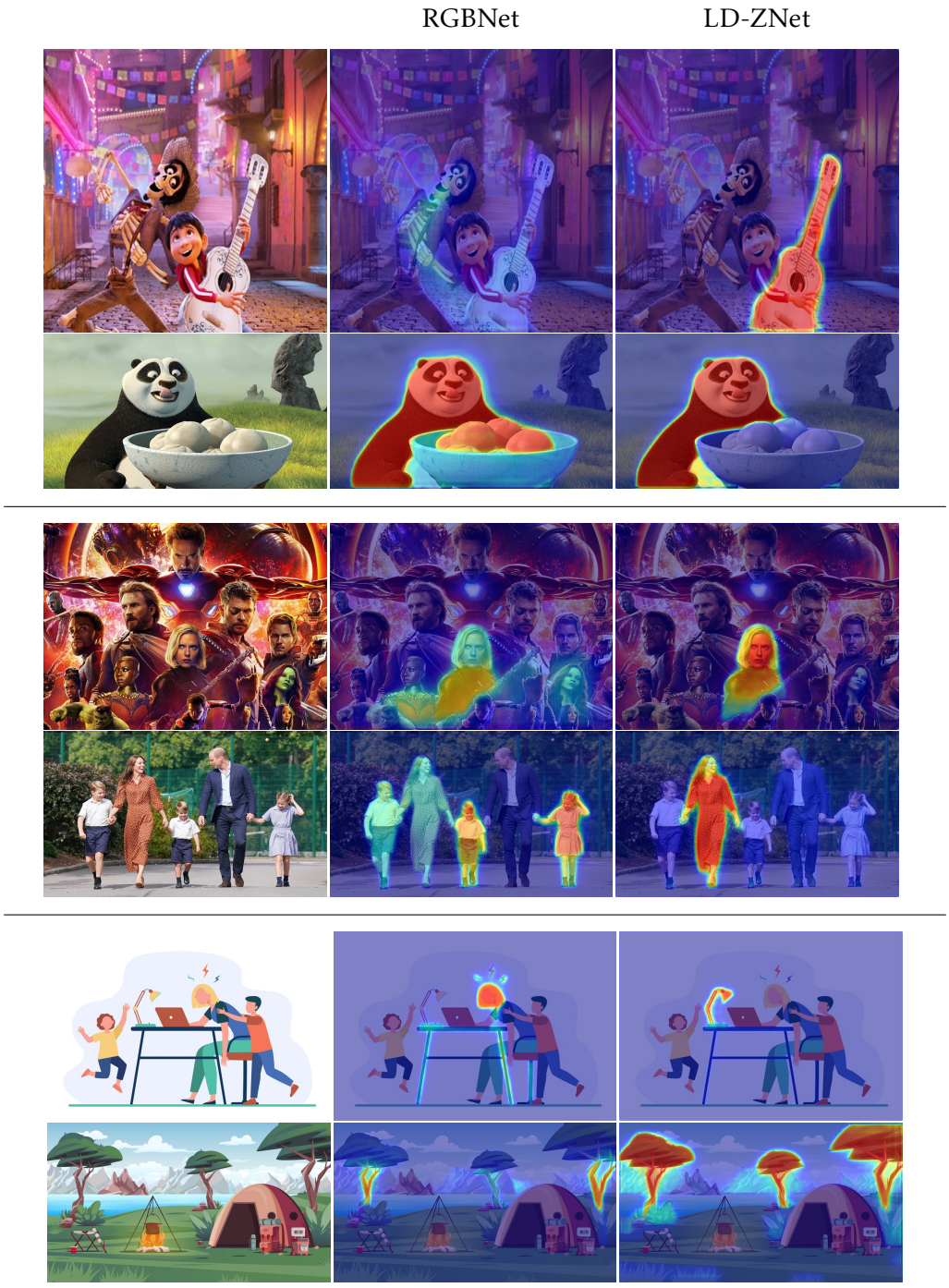


Figure 3.12: More qualitative examples where RGBNet fails to localize *"Guitar"*, *"Panda"* from animation images (top row), famous celebrities *"Scarlett Johansson"*, *"Kate Middleton"* (second row) and objects such as *"Lamp"*, *"Trees"* from illustrations (bottom row). LD-ZNet benefits from using z combined with the internal LDM features to correctly segment these text prompts.

Chapter 4: Conclusions and Future Work

4.1 Concluding Remarks

In this dissertation, we presented novel ways to utilize two popular deep generative models namely GANs and Diffusion models to improve crucial tasks in computer vision - 1) Geometry Estimation and 2) Text-Based Image Segmentation, respectively.

1. *GANs for Unsupervised Geometry Estimation*. In Chapter 2, we proposed a generative-based SharinGAN module for unsupervised domain adaptation (UDA) to combine labeled synthetic and unlabeled real images during training. The SharinGAN translates just the domain-specific task-related information from both domains into a shared space that is input to the primary task network. The information unrelated to the task is untouched by SharinGAN during this translation for both domains. With this formulation, we show a much improved generalization of the primary task network on various estimation tasks - Monocular Depth Estimation of outdoor scenes, Face Normal Estimation, and Lighting Estimation, all in an unsupervised setting.

2. *LDMs for Text-Based Image Segmentation*. In Chapter 3, we proposed to use large-scale latent diffusion models (LDM) pretrained on the internet to improve text-based segmentation performance for several novel classes from the internet and on a variety of imagery - Real, AI-generated, illustrations, animations etc. The understanding of internet-scale concepts along with the ability to synthesize various photorealistic objects from text, makes the LDM an intuitive candidate to improve text-based recognition performance. Our proposed segmentation pipeline LD-ZNet benefits from the z-space as well as the internal representations within LDM that is shown to contain semantic information. We showed improved segmentation performance for LD-ZNet on not just real images but also on AI-Generated images, animations, illustrations and celebrity images etc.

4.2 Future Work

As we move towards an era of large-scale datasets with higher compute, the generative models trained with them can only get more powerful. It thus becomes crucial to understand how to utilize these generative models to improve general computer vision systems.

In Chapter 2, we tackled the UDA problem by translating the labeled synthetic and unlabeled real images into a shared space via a GAN model. Alternately, going forward, we can rely on much more powerful generative models to render more photorealistic images along with their labels, reducing the gap to the

real domain. The internal representations of the generative models could contain powerful semantic information to support such paired synthesis of images and corresponding labels such as depth, normals, segmentation maps etc.

In Chapter 3, we relied on the semantic knowledge of objects from the internet present inside a pretrained LDM, to improve the text-based segmentation across novel concepts such as Mickey Mouse, Spiderman etc., without requiring any sort of annotations for these concepts. This opens a new way to bypass the limitations in existing datasets that often have a fixed number of object categories. For example, we can rely on a pretrained LDM that can synthesize photorealistic MRI images from text, to improve segmentation performance of different lesions. Moreover the lateset advances in generative works on novel-view synthesis [133], text-to-video [134–139] and text-to-3D [140, 141] are based on the pretrained diffusion models which suggest they contain more knowledge that can potentially improve several downstream applications.

Also in Chapter 3, we described the recent traction of AI-Generated images due to the growing popularity and availability of more powerful generative models. Several editing workflows such as inpainting are applied on the AI-Generated images and thus it becomes necessary to develop computer vision systems that generalize well to the AI-Generated content. We released a dataset named AIGI that contains 100 AI-Generated images along with object labels and categorical captions, to help evaluate the generalization capabilities of existing text-based image segmentation methods. Going forward, such AI-datasets are necessary for various tasks such as object detection, geometry estimation etc.

Appendix A: Bidirectional Convolutional LSTM for the Detection of Violence in Videos

[1]In recent years, the problem of human action recognition from video has gained momentum in the field of computer vision [143–145]. Despite its usefulness, the specific task of violence detection has been comparatively less studied. However, violence detection has huge applicability in public security and surveillance markets. Surveillance cameras are deployed in large numbers, particularly in schools, prisons etc. Problems such as lack of personnel and slow response arise, leading to a strong demand for automated violence detection systems. Additionally, with the surge in easy-to-access data uploaded to social media sites and across the web, it is imperative to develop automated methods to childproof the internet. Hence, in recent years, focus has been directed towards solving this problem [146–150].

[1]Work done with Alex Hanson, Sanjukta Krishnagopal, and Larry Davis. Accepted [142] in ECCV 2018 Workshops. This is placed in the appendix because it is an early thesis work that does not directly connect to the main content of this dissertation.

A.1 Contributions and Proposed Approach

In this work, we propose a Bidirectional Convolutional LSTM (BiConvLSTM) [151–153] architecture, called the Spatiotemporal Encoder, to detect violence in videos. Our architecture builds on existing ConvLTSM architectures in which we include a bidirectional temporal encoding and elementwise max pooling, novel in the field of violence detection. We encode each video frame as a collection of feature maps via a forward pass through a VGG13 network [154]. We then pass these feature maps to a BiConvLSTM to further encode them along the video's temporal direction, performing both a pass forward in time and in reverse. Next, we perform an elementwise maximization on each of these encodings to create a representation of the entire video. Finally, we pass this representation to a classifier to identify whether the video contains violence. This extends the architecture of [146], which uses a Convolutional LSTM (ConvLSTM) by encoding temporal information in both directions. We speculate that access to both future and past inputs from a current state allows the BiConvLSTM to understand the context of the current input, allowing for better classification on heterogeneous and complex datasets. We validate the effectiveness of our networks by running experiments on three standard benchmark datasets commonly used for violence detection, namely, the Hockey Fights dataset (HF), the Movies dataset (M), and the Violent Flows dataset (VF). We find that our architecture matches state-of-the-art on the Hockey Fights [155] and Movies [155] datasets and performs comparably with other methods on the Violent Flows [156] dataset. Surprisingly, a

simplified version of our architecture, called the Spatial Encoder, also matches state-of-the-art on Hockey Fights and Movies, leading us to speculate that these datasets may be comparatively smaller and/or simpler for the task of violence detection.

This paper is outlined as follows. Section 2 provides more detail about the model architectures we propose. Section 3 describes the datasets used in this work. Section 4 summarizes the training methodology. And section 5 presents our experimental results and ablation studies.

A.2 Related Work

Early work in the field includes [157], where violent scenes in videos were recognized by using flame and blood detection and capturing the degree of motion, as well as the characteristic sounds of violent events. Significant work has been done on harnessing both audio and video features of a video in order to detect and localize violence [158]. For instance, in [159], a weakly supervised method is used to combine auditory and visual classifiers in a co-training way. While incorporating audio in the analysis may often be more effective, audio is not often available in public surveillance videos. We address this problem by developing an architecture for violence detection that does not require audio features.

Additionally, violence is a rather broad category, encompassing not only person-person violence, but also crowd violence, sports violence, fire, gunshots,

physical violence etc. In [160], crowd violence is detected using Latent Dirichlet Allocation (LDA) and Support Vector Machines (SVMs). Violence detection through specific violence-related object detection such as guns is also a current topic of research [161].

Several existing techniques use inter-frame changes for violence detection, in order to capture fast motion changing patterns that are typical of violent activity. [162] proposed the use of acceleration estimates computed from the power spectrum of adjacent frames as an indicator of fast motion between successive frames. [146] proposed a deep neural network for violence detection by feeding in frame differences. [163] proposed using blob features, obtained by subtracting adjacent frames, as the feature descriptor.

Other methods follow techniques such as motion tracking and position of limbs etc. to identify spatiotemporal interest points and extract features from these points. These include Harris corner detector [164], Motion Scale-Invariant Feature Transform (MoSIFT) [165]. MoSIFT descriptors are obtained from salient points in two parts: the first is an aggregated Histogram of Gradients (HoG) which describe the spatial appearance. The second part is an aggregated Histogram of optical Flow (HoF) which indicates the movement of the feature point. [147] used a modified version of motion-Weber local descriptor (MoIWLD), followed by sparse representation as the feature descriptor.

Additional work has used the Long Short-Term Memory (LSTM) [166] deep learning architecture to capture spatiotemporal features. [148] used LSTMs for feature aggregation for violence detection. The method consisted of extracting

features from raw pixels using a CNN, optical flow images and acceleration flow maps followed by LSTM based encoding and a late fusion. Recently, [167] replaced the fully-connected gate layers of the LSTM with convolutional layers and used this improved model (named ConvLSTM) for predicting precipitation nowcasting from radar images with improved performance. This ConvLSTM architecture was also successfully used for anomaly prediction [168] and weakly-supervised semantic segmentation in videos [169].

Bidirectional RNNs are first introduced in [170]. Later, [171] proposed using the same for speech recognition task and was shown to perform better than an unidirectional RNN. Recently, bidirectional LSTMs were used in predicting network-wide traffic speed [172], framewise phoneme classification [173] etc. showing they are better in terms of prediction than unidirectional LSTMs. The same concept has been leveraged for tasks involving videos such as video-super resolution [174], object segmentation in a video [175] and learning spatiotemporal features for gesture recognition [147] and fine-grained action detection [153]. While several of these incorporate a convolutional module coupled with an RNN module, our architecture extends this by the inclusion of temporal encoding in both forward and backward temporal directions, through the use of a BiConvLSTM and elementwise max pooling. We speculate that the access of future information from the current state is particularly beneficial in more heterogenous datasets.

A.3 Model Architecture

To appropriately classify violence in videos we sought to generate a robust video encoding to pass through a fully connected classifier network. We produce this video representation through a spatiotemporal encoder. This extracts features from a video that correspond to both spatial and temporal details via a Spatiotemporal Encoder (Sec. A.3.1). The temporal encoding is done in both temporal directions, allowing access to future information from the current state. We also study a simplified version of the spatiotemporal encoder that encodes only spatial features via a simplified Spatial Encoder (Sec. A.3.2). The architectures for both encoders are described below.

A.3.1 Spatiotemporal Encoder Architecture

The Spatiotemporal Encoder architecture is shown in Fig. A.1. It consists of a spatial encoder that extracts spatial features for each frame in the video followed by a temporal encoder that allows these spatial feature maps to 'mix' temporally to produce a spatiotemporal encoding at each time step. All of these encodings are then aggregated into a single video representation via an elementwise max pooling operation. This final video representation is vectorized and passed to a fully connected classifier.

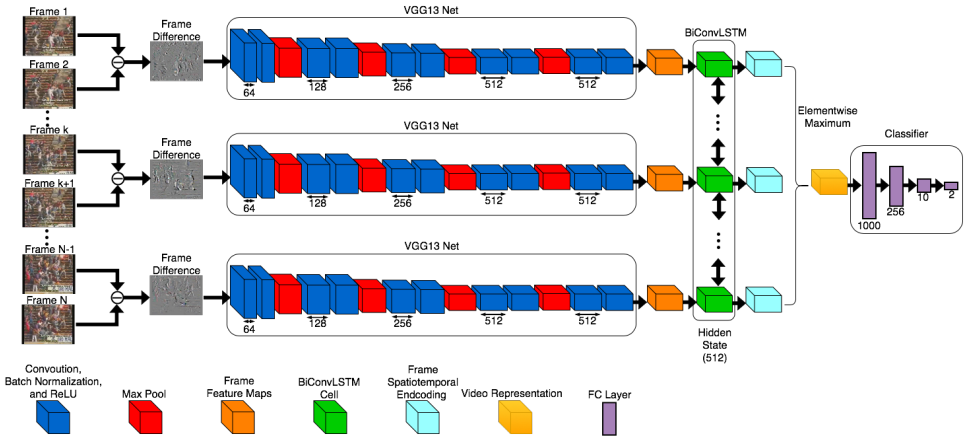


Figure A.1: The Spatiotemporal Encoder is comprised of three parts: a VGG13 network spatial encoder, a Bidirectional Convolution LSTM (BiConvLSTM), temporal encoder, and a classifier. Frames are resized to 224×224 and the difference between adjacent frames is used as input to the network. The VGG classifier and last max pooling layer is removed from VGG13 network (Blue and Red). The frame feature maps (Orange), are size $14 \times 14 \times 512$. The frame features are passed to the BiConvLSTM (Green) which outputs the frame spatiotemporal encodings (Cyan). An elementwise max pooling operation is performed on the spatiotemporal encoding to produce the final video representation (Gold). This video representation is then classified as violent or nonviolent via a fully connected classifier (Purple).

A.3.1.1 Spatial Encoding

In this work, a VGG13 [154] convolutional neural network (CNN) model is used as the spatial encoder. The last max pool layer and all fully connected layers of the VGG13 net are removed, resulting in spatial feature maps for each frame of size $14 \times 14 \times 512$. Instead of passing video frames directly, adjacent frames were subtracted and used as input to the spatial encoder. This acts as pseudo-optical

flow model and follows [146, 176].

A.3.1.2 Temporal Encoding

A Bidirectional Convolutional LSTM (BiConvLSTM) is used as the temporal encoder, the input to which are the feature maps from the spatial encoder. We constructed the BiConvLSTM in such a way that the output from each cell is also $14 \times 14 \times 512$. The elementwise maximum operation is applied to these outputs as depicted in Fig. A.1, thus resulting in a final video representation of size $14 \times 14 \times 512$.

A BiConvLSTM cell is essentially a ConvLSTM cell with two cell states. We present the functionality of ConvLSTM and BiConvLSTM in the following subsections.

ConvLSTM: A ConvLSTM layer learns global, long-term spatiotemporal features of a video without shrinking the spatial size of the intermediate representations. This encoding takes place during the recurrent process of the LSTM. In a standard LSTM network the input is vectorized and encoded through fully connected layers, the output of which is a learned temporal representation. As a result of these fully connected layers, spatial information is lost. Hence, if one desires to retain that spatial information, the use of a convolutional operation instead of fully connected operation may be desired. The ConvLSTM does just that. It replaces the fully connected layers in the LSTM with convolutional layers. The ConvLSTM is utilized in our work such that the convolution and recurrence

operations in the input-to-state and state-to-state transitions can make full use of the spatiotemporal correlation information. The formulation of the ConvLSTM cell is shown below:

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + b_i)$$

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + b_f)$$

$$o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + b_o)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c)$$

$$H_t = o_t \odot tanh(C_t)$$

Where "*" denote the convolution operator, "o" denote the Hadamard product, "σ" is the sigmoid function and W_{x*}, W_{h*} are 2D Convolution kernels that corresponding to the input and hidden state respectively. The hidden $(H_0, H_1, ..H_{t-1})$ and the cell states $(C_1, C_2, ..C_t)$ are updated based on the input $(X_1, X_2, ..X_t)$ that pass through i_t, f_t and o_t gate activations during each time sequence step. b_i, b_f, b_o and b_c are the corresponding bias terms.

BiConvLSTM: The BiConvLSTM is an enhancement to ConvLSTM in which two sets of hidden and cell states are maintained for each LSTM cell: one for a forward sequence and the other for a backward sequence in time. BiConvLSTM can thereby access long-range context in both directions of the time sequence of the input and thus potentially gain a better understanding of the entire video. Fig. A.2 illustrates the functionality of a BiConvLSTM Cell. It is comprised of a ConvLSTM cell with two sets of hidden and cell states. The first set (h_f, c_f)

is for forward pass and the second set (h_b, c_b) is for backward pass. For each time sequence, the corresponding hidden states from the two sets are stacked and passed through a Convolution layer to get a final hidden representation for that time step. That hidden representation is then passed to the next layer in the BiConvLSTM module as input.

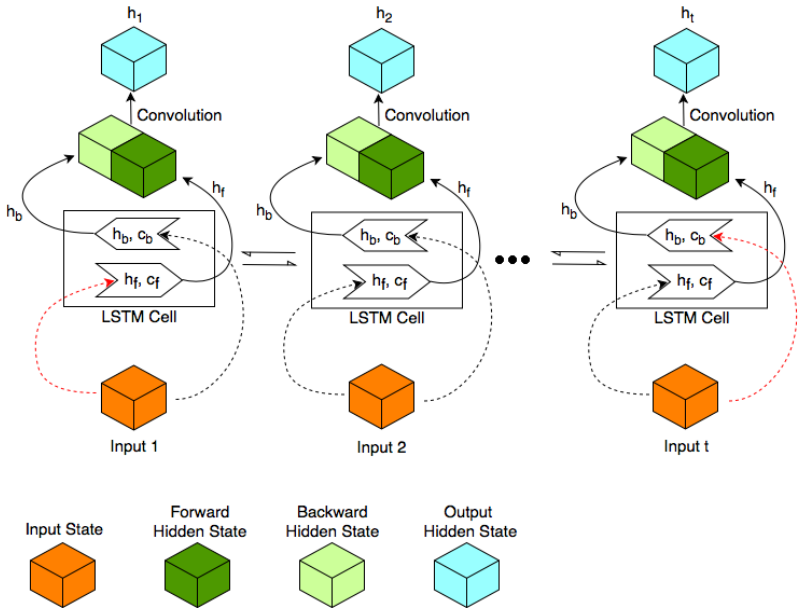


Figure A.2: Overview of a BiConvLSTM Cell. The hidden and cell states are passed to the next LSTM cell in the direction of flow. Red dashed lines correspond to the first input in the time step for both the forward and backward hidden states.

A.3.1.3 Classifier

The number of nodes in each layer in the fully connected classifier, ordered sequentially, are 1000, 256, 10, and 2. Each layer utilizes the hyperbolic tangent non-linearity. The output of the last layer is a binary predictor into classes violent and non-violent .

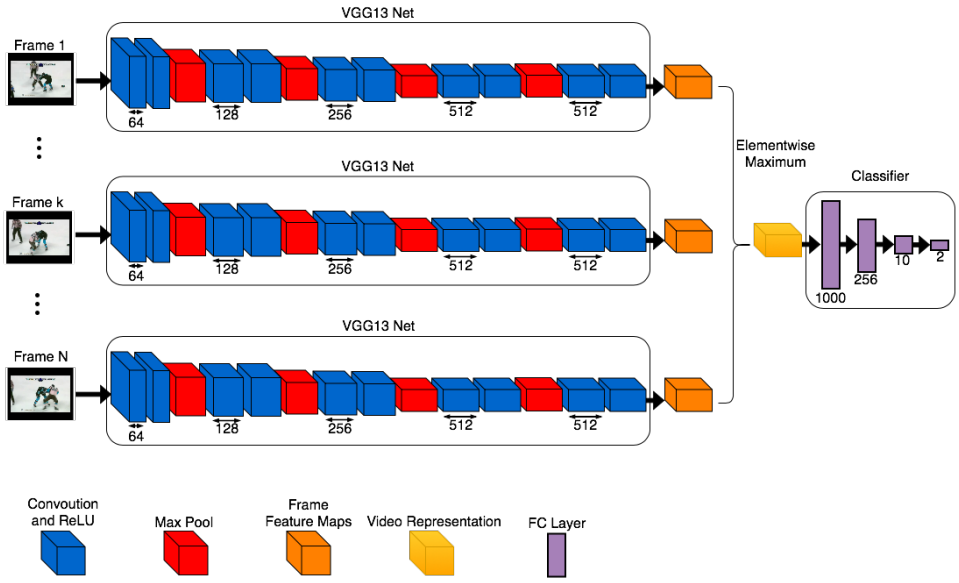


Figure A.3: The Spatial Encoder is comprised of two parts: a VGG13 network spatial encoder and a classifier. Frames are resized to 224×224 before provided as input to the network. The VGG classifier and last max pooling layer are removed from VGG13 network (Blue and Red). The frame feature maps (Orange), are size $14 \times 14 \times 512$. An elementwise max pooling operation is performed on the frame feature maps to produce the final video representation (Gold). This video representation is then classified as violent or nonviolent via a fully connected classifier (Purple).

A.3.2 Spatial Encoder Architecture

Spatial Encoder is a simplified version of the Spatiotemporal Encoder architecture (Sec A.3.1) and is shown in Fig. A.3. The temporal encoder is removed and elementwise max pooling is applied directly to the spatial features. Additionally, since we are interested in purely the spatial features in this architecture, adjacent frame differences are not used as input and frames are passed directly to the spatial encoder.

A.4 Data

Details about the three standard datasets widely used in this work are provided below. For all datasets, we downsampled each video to 20 evenly spaced frames as input to the network.

Hockey Fights dataset (HF) was created by collecting videos of ice hockey matches and contains 500 fighting and non-fighting videos. Almost all the videos in the dataset have a similar background and subjects (humans).

Movies dataset (M) consists of fight sequences collected from movies. The non-fight sequences are collected from publicly available action recognition datasets. The dataset is made up of 100 fight and 100 non-fight videos. As opposed to the hockey fights dataset, the videos of the movies dataset are substantially different in their content.

Violent Flows dataset (VF) is a database of real-world, video footage of crowd violence, along with standard benchmark protocols designed to test both violent/non-violent classification and violence outbreak detection. The data set contains 246 videos. All the videos were downloaded from YouTube. The shortest clip duration is 1.04 seconds, the longest clip is 6.52 seconds, and the average length of a video clip is 3.60 seconds.

A.5 Training Methodology

For the spatial encoder, the weights were initialized as the pretrained ImageNet [177] weights for VGG13. For the Spatiotemporal Encoder, the weights of the BiConvLSTM cell and classifier were randomly initialized. Frame differences were taken for the Spatiotemporal Encoder architecture and frames were normalized to be in the range of 0 to 1. For both architectures, the learning rate was chosen to be 10^{-6}. A batch size of 8 video clips were used as input and the weight decay was set to 0.1. ADAM optimizer with default beta range $(0.5, 0.99)$ was used. Frames were selected at regular intervals and resized to 224×224. Additionally, random cropping (RC) and random horizontal flipping (RHF) data augmentations were used for the Hockey Fights and Movies clips, where as only RHF was applied to Violent Flows clips. Cross entropy loss was used during training. Furthermore, 5-fold cross validation was used to calculate performance.

A.6 Results

The following subsections (A.6.1 and A.6.2) discuss the results and the corresponding model that obtained best performance for all three datasets.

A.6.1 Hockey Fights and Movies

The best performance for the Hockey Fights and Movies datasets was observed with the simpler Spatial Encoder Architecture depicted in Fig. A.3 and

described in Section A.3.2. We obtained an accuracy of 96.96±1.08% on the Hockey Fights dataset and an accuracy of 100±0% on the Movies dataset, both of which match state-of-the-art. A comparison of our results with other recent work is given in Table A.1. While our model performance was saturated at 100 ± 0% for the Movies dataset, it outperformed previous methods with comparable accuracy measures (Table A.1 rows 1-11) by a statistically significant margin and hence, we believe, is a significant improvement.

These results, in contrast to most prior work, were attained without the use of a temporal encoding of the features. While the Spatiotemporal Encoder performed comparably to the Spatial Encoder, we observe that the additional level of complexity involved in utilizing the temporal features wasn't justified for datasets like Movies and Hockey Fight that are relatively more homogeneous than the Violent Flows dataset. We speculate that for certain domains, robust spatial information may be sufficient for violence classification.

A.6.2 Violent Flows

The best performance on the Violent Flows dataset was observed using the Spatiotemporal Encoder architecture shown in Fig. A.1 and described in Section A.3.1. Our accuracy on the Violent Flows dataset was 92.18±3.29%. While not state-of-the-art, this accuracy is comparable to existing recent methods as shown in Table A.1. We noticed batch normalization caused a decrease in performance on the Violent Flows dataset. Hence, all reported accuracies for the Violent Flows

dataset were obtained without applying batch normalization in the networks.

A.6.3 Accuracy Evaluation

Due to the small size of the datasets, we chose to employ 5-fold cross validation to evaluate model accuracies. We split each dataset into 5 equal sized and randomly partitioned folds. One fold is reserved for testing and the other four are used for training. The model is trained from scratch once for each test fold and hence five test accuracies are obtained per epoch of training. We calculate the mean per epoch of these accuracies and locate the epoch with maximal accuracy value. We then calculate the mean and standard deviation of all 100 test accuracies that lie within a 10 epoch radius of this maximal accuracy. We report this as our overall model accuracy and standard deviation.

This contrasts the accuracy evaluation used in [146], where for each fold the maximum value over all epochs is obtained, and the mean of these values is reported [181]. For completeness, we report our accuracies using this evaluation method in Table A.1 using a '*'.

As shown in Fig. A.4, the mean test accuracy for the Hockey Fights dataset peaks at 97.3% for epoch 63. We take the mean and standard deviation of test accuracies from epoch 53 to epoch 73 and obtain an overall accuracy of $96.96 \pm 1.08\%$.

Fig. A.5 shows the mean test accuracy of the Violent Flows dataset to be 94.69 at epoch 710. The mean and standard deviation of test accuracies between

Table A.1: Performance comparison of different methods for Hockey Fights, Movies, and Violent Flows datasets. In the Hockey and Movies datasets our proposed methods match the state-of-the-art performance. In the case of the Violent Flows dataset, our method is comparable to existing methods. The best performance for each dataset and our proposed methods are highlighted in bold. Two methods for calculating accuracies are used here. Accuracy calculation of rows 1 – 11 are outlined in Sec. A.6.3.

**For the purpose of fair comparison with [146], we also present performance measured through the accuracy calculation of [146]. For more details refer to Sec. A.6.3*

Method	**Hockey**	**Movies**	**Violent Flows**
MoSIFT+HIK [155]	90.9%	89.5%	-
ViF [156]	82.9±0.14%	-	81.3±0.21%
MoSIFT+KDE+Sparse Coding [178]	94.3±1.68%	-	89.05±3.26%
Deniz et al. [179]	90.1±0%	98.0±0.22%	-
Gracia et al. [163]	82.4±0.4%	97.8±0.4%	-
Substantial Derivative [180]	-	96.89±0.21%	85.43±0.21%
Bilinski et al. [149]	93.4%	99%	**96.4%**
MoIWLD [147]	**96.8±1.04%**	-	93.19±0.12%
ViF+OViF [150]	87.5±1.7%	-	88±2.45%
Three streams + LSTM [148]	93.9	-	-
Proposed: Spatiotemporal Encoder	**96.54±1.01%**	**100±0%**	92.18±3.29%
Proposed: Spatial Encoder	**96.96±1.08%**	**100±0%**	90.63±2.82%
Swathikiran et al. [146]	97.1±0.55%*	**100±0%***	94.57±2.34%*
Proposed: Spatiotemporal Encoder	97.9±0.37%*	**100±0%***	**96.32±1.52%***
Proposed: Spatial Encoder	**98.1±0.58%***	**100±0%***	93.87±2.58%*

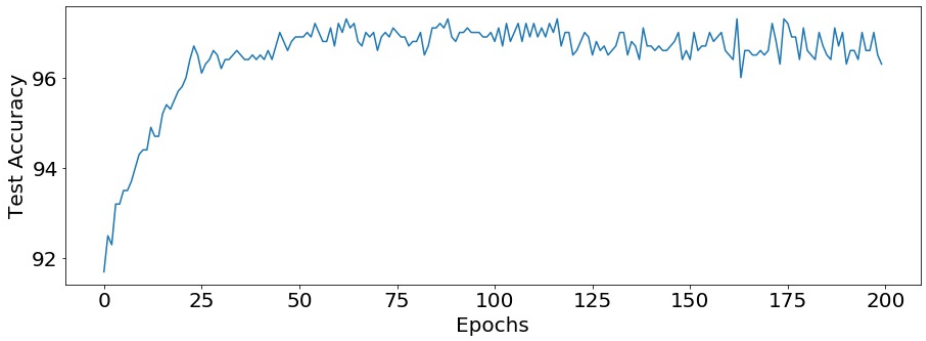


Figure A.4: Mean fold accuracy on Hockey evaluated using the Spatial Encoder architecture.

epoch 700 and 720 produces an overall accuracy of $92.18 \pm 3.29\%$.

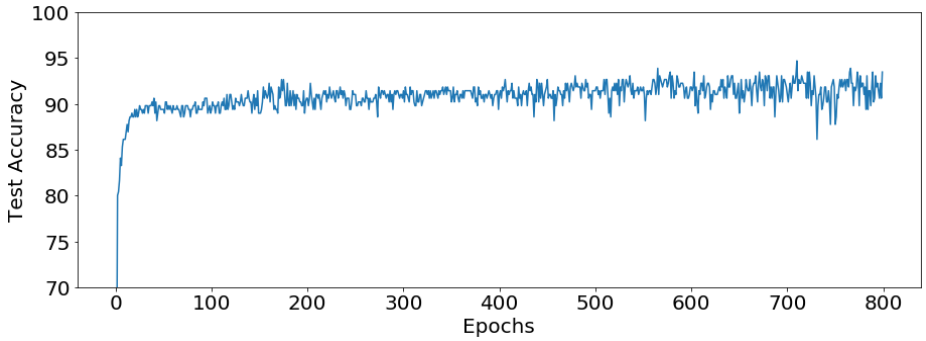


Figure A.5: Mean fold accuracy on Violent Flows evaluated using the Spatiotemporal Encoder architecture.

The Movies dataset converged to 100.0% after 3 epochs. Hence we report an overall accuracy of 100.0% for this dataset.

A.6.4 Ablation Studies

We conducted several ablation studies to determine how the boost in performance can be attributed to the key components in our Spatiotemporal Encoder Architecture. In particular, we examine the effects of using a VGG13 network

pretrained on ImageNet to encode spatial features, the use of a BiConvLSTM network to refine these encodings temporally, and the use of elementwise max pooling to create an aggregate video representation. To baseline performance gains, we compare against architectural decisions made by the study that most closely resembles our work, [146].

A.6.4.1 Spatial vs Spatiotemporal Encoders

This study examines the role of a temporal encoder during classification. The performance of the Spatial (Sec. A.3.2) and Spatiotemporal (Sec. A.3.1) Encoders are compared and illustrated in Fig. A.6 and Fig. A.7 for the Hockey and Violent Flows respectively. We see the temporal encoding is adding a slight boost in performance in the case of Violent Flows. However, the simpler Spatial Encoder architecture performs slightly better for the hockey dataset.

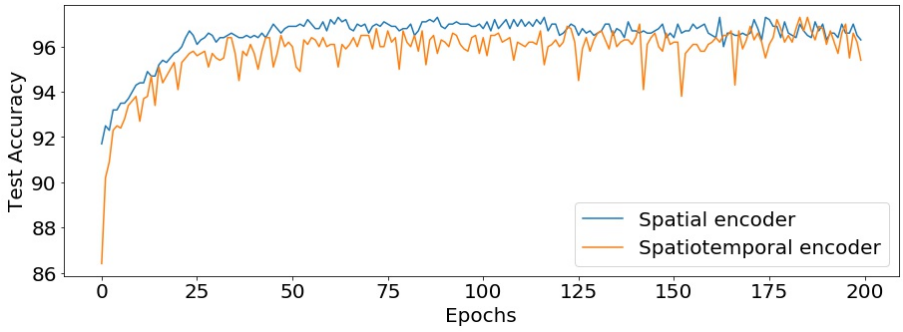


Figure A.6: Performance comparison between spatial and spatiotemporal encoders on the Hockey dataset.

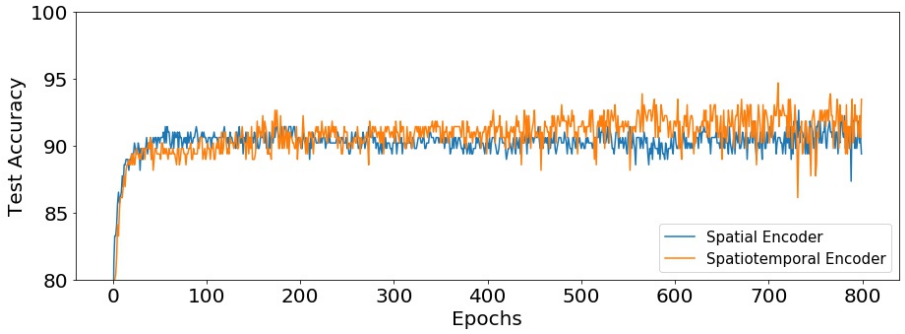


Figure A.7: Performance comparison between spatial and spatiotemporal encoders on the Violent Flows dataset.

A.6.4.2 Elementwise Max Pooling vs. Last Encoding

In this study, we sought to determine the usefulness of aggregating the spatiotemporal encodings via the elementwise max pool operation. We did so by removing the elementwise max pooling operation and running classification on the last spatiotemporal frame representation. Fig. A.8 depicts that using elementwise max pool aggregation lead to significant improvement in performance.

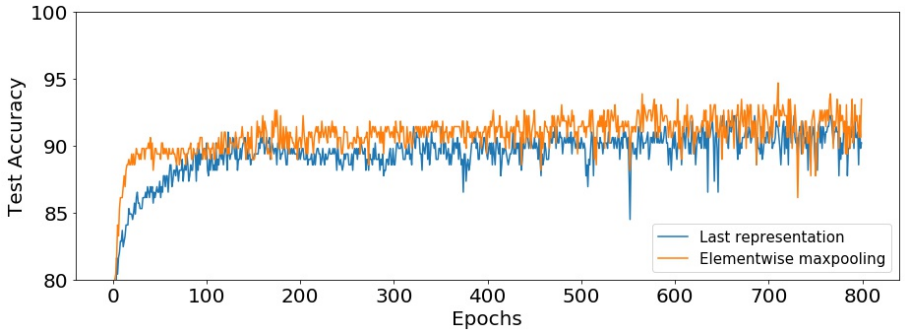


Figure A.8: Performance comparison between the feature aggregation techniques max pooling and last time sequence representation from the BiConvLSTM module on the Violent Flows dataset.

A.6.4.3 ConvLSTM vs. BiConvLSTM

For this study, we evaluated the impact of bidirectionality of the BiConvLSTM on violence classification. We compared its performance to a ConvLSTM module and depict the accuracies of both in Fig. A.9. BiConvLSTM yields a slightly higher classification accuracy.

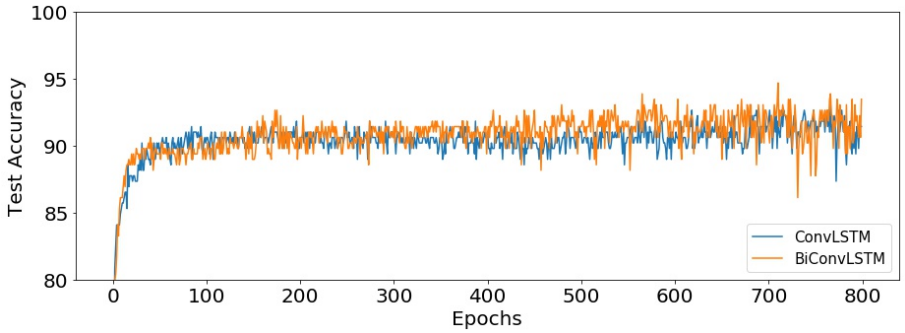


Figure A.9: Performance comparison between ConvLSTM and BiConvLSTM as temporal encoders on the Violent Flows dataset.

A.6.4.4 AlexNet vs. VGG13

The aim of this study was to understand the affect of different spatial encoder architectures on the classification performance. For this we chose AlexNet and VGG13 Net pretrained on ImageNet as spatial encoders. Fig.A.10 shows the performance comparison for the two encoders. It is apparent that VGG13 is performing appreciably better than AlexNet.

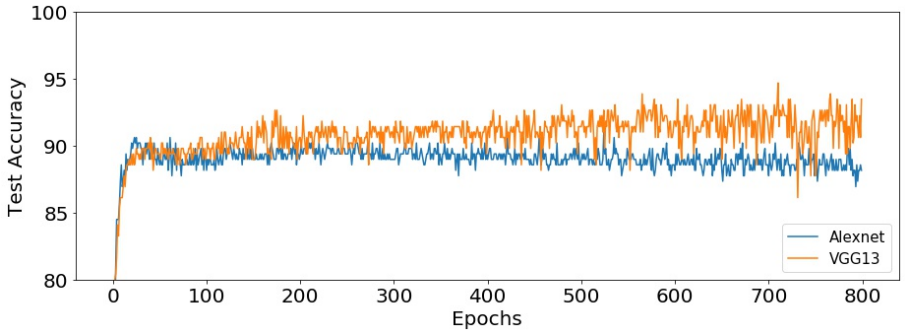


Figure A.10: Performance comparison between AlexNet and VGG13 pretrained models as spatial encoders on Violent Flows dataset.

A.7 Conclusions

We have proposed a Spatiotemporal Encoder architecture and a simplified Spatial Encoder for supervised violence detection. The former performs reasonably well on all the three benchmark datasets whereas the later matches state-of-the-art performance on the Hockey Fights and Movies datasets. We presented various ablation studies that demonstrate the significance of each module in the spatiotemporal encoder model and provide grounding for our architectures.

While several studies have used ConvLSTMs for video related problems, our contribution of introducing bidirectional temporal encodings and the elementwise max pooling of those encodings facilitates better context-based representations. Hence, our Bidirectional ConvLSTM performs better for more heterogeneous and complex datasets such as the Violent Flows dataset compared to the ConvLSTM architecture [146]. Based on the comparisons in the results section, it is not clear if there is a method that is consistently best. Current commonly

used benchmark violence datasets are relatively small (a few hundred videos) compared to traditional deep learning dataset sizes. We anticipate that larger datasets may lead to better comparisons between methods. This may constitute an interesting future course of study.

Additionally, we were surprised by the performance of the Spatial Encoder Architecture. Violence detection is a hard problem, but we speculate that some datasets may be easier than others. Pause a movie or hockey match at just the right frame and it is likely that a human user will be able to tell if a fight scene or brawl is taking place. We hypothesize that the same is true for a neural network. A specific frame may fully encode violence in a video for a particular domain. We speculate that this is why our Spatial Encoder Architecture was able to match state-of-the-art on the Hockey Fights and Movies datasets. For more complex datasets and scenes with rapidly changing violence features, it is important to understand the context of the frame in the whole video, i.e., both the past video trajectory and future video trajectory leading outwards from that frame. This is particularly true for longer or more dynamic videos with greater heterogeneity; the same sequence of frames could go one of several directions in the future. It is for this reason that we believe our novel contributions to the architecture, the 'Bi' in the BiConvLSTM and elementwise max pooling, are beneficial to develop better video representations, and we speculate that our architecture may perform well on more dynamic and heterogeneous datasets. We anticipate further investigation into this may lead to fruitful results.

Bibliography

[1] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. *Advances in neural information processing systems*, 32, 2019.

[2] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020.

[3] Nontawat Tritrong, Pitchaporn Rewatbowornwong, and Supasorn Suwajanakorn. Repurposing gans for one-shot semantic part segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4475–4485, 2021.

[4] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10145–10155, 2021.

[5] Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. *arXiv preprint arXiv:2112.03126*, 2021.

[6] Jianjin Xu and Changxi Zheng. Linear semantics in generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9351–9360, 2021.

[7] Yinghao Xu, Yujun Shen, Jiapeng Zhu, Ceyuan Yang, and Bolei Zhou. Generative hierarchical features from synthesizing images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4432–4442, 2021.

[8] Jeevan Devaranjan, Sanja Fidler, and Amlan Kar. Unsupervised learning of scene structure for synthetic data generation, September 9 2021. US Patent App. 17/117,425.

[9] Amlan Kar, Aayush Prakash, Ming-Yu Liu, Eric Cameracci, Justin Yuan, Matt Rusiniak, David Acuna, Antonio Torralba, and Sanja Fidler. Meta-sim: Learning to generate synthetic datasets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4551–4560, 2019.

[10] Daiqing Li, Amlan Kar, Nishant Ravikumar, Alejandro F Frangi, and Sanja Fidler. Federated simulation for medical imaging. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 159–168. Springer, 2020.

[11] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[12] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[13] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *NeurIPS*. 2016.

[14] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016.

[15] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation, 2017.

[16] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[17] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc., 2021.

[18] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[19] Koutilya PNVR, Hao Zhou, and David Jacobs. Sharingan: Combining synthetic and real data for unsupervised geometry estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[20] Kevin Karsch, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, Hailin Jin, Rafael Fonte, Michael Sittig, and David Forsyth. Automatic scene inference for 3d object compositing. *ToG*, 33(3), 2014.

[21] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724, 2015.

[22] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D. Castillo, and David W. Jacobs. Sfsnet: Learning shape, refectance and illuminance of faces in the wild. In *CVPR*, 2018.

[23] Y. Wang, L. Zhang, Z. Liu, G. Hua, Z. Wen, Z. Zhang, and D. Samaras. Face relighting from a single image under arbitrary unknown lighting conditions. *IEEE Trans. on PAMI*, 31(11):1968–1984, 2009.

[24] Hao Zhou, Sunil Hadap, Kalyan Sunkavalli, and David W. Jacobs. Deep single portrait image relighting. In *ICCV*, 2019.

[25] Jogendra Nath Kundu, Phani Krishna Uppala, Anuj Pahuja, and R. Venkatesh Babu. Adadepth: Unsupervised content congruent adaptation for depth estimation. In *CVPR*, 2018.

[26] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In *ECCV*, 2018.

[27] Shanshan Zhao, Huan Fu, Mingming Gong, and Dacheng Tao. Geometry-aware symmetric domain adaptation for monocular depth estimation. In *CVPR*, 2019.

[28] Amir Atapour-Abarghouei and Toby P. Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *CVPR*, June 2018.

[29] Yohann Cabon Eleonora Vig Adrien Gaidon, Qiao Wang. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016.

[30] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015.

[31] Ashutosh Saxena, Min Sun, and Andrew Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Trans. PAMI*, 31(5):824–840, 2009.

[32] Stefanos Zafeiriou, Mark F. Hansen, Gary A. Atkinson, Vasileios Argyriou, Maria Petrou, Melvyn L. Smith, and Lyndon N. Smith. The photoface database. In *CVPR Workshops*, 2011.

[33] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng. Learning depth from single monocular images. In *NeurIPS*, 2006.

[34] Miaomiao Liu, Mathieu Salzmann, and Xuming He. Discrete-continuous depth estimation from a single image. In *CVPR*, June 2014.

[35] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*. 2014.

[36] David Eigen, , and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015.

[37] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Trans. on PAMI*, 38(10), 2016.

[38] Lei He, Guanghui Wang, and Zhanyi Hu. Learning depth from single images with deep neural network embedding focal length. *IEEE Trans. on Image Processing*, 27(9), 2018.

[39] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *CVPR*, 2018.

[40] Vamshi Repala and Shiv Ram Dubey. Dual cnn models for unsupervised monocular depth estimation. 04 2018.

[41] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *CVPR*, 2018.

[42] Anirban Roy and Sinisa Todorovic. Monocular depth estimation using neural regression forest. In *CVPR*, 2016.

[43] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.

[44] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017.

[45] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *CVPR*, 2018.

[46] Vincent Casser, Sören Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *AAAI*, 2019.

[47] Clement Godard, Mac Aodha Oisin, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, 2019.

[48] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

[49] Ishit Mehta, Parikshit Sakurikar, and P. Narayanan. Structured adversarial training for unsupervised monocular depth estimation. In *3DV*, 2018.

[50] Matteo Poggi, Fabio Tosi, and Stefano Mattoccia. Learning monocular depth estimation with unsupervised trinocular assumptions. In *3DV*, 2018.

[51] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *ICRA*, 2019.

[52] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, 1999.

[53] Ayush Tewari, Michael Zollöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Theobalt Christian. MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction. In *ICCV*, 2017.

[54] Zhixin Shu, Ersin Yumer, Sunil Hadap, Kalyan Sunkavalli, Eli Shechtman, and Dimitris Samaras. Neural face editing with intrinsic image disentangling. In *CVPR*, 2017.

[55] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T. Freeman. Unsupervised training for 3d morphable model regression. *CVPR*, 2018.

[56] Luan Tran and Xiaoming Liu. Nonlinear 3d face morphable model. In *CVPR*, 2018.

[57] Luan Tran, Feng Liu, and Xiaoming Liu. Towards high-fidelity nonlinear 3d face morphoable model. In *CVPR*, 2019.

[58] Feng Liu, Luan Tran, and Xiaoming Liu. 3d face modeling from diverse raw scan data. In *ICCV*, 2019.

[59] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*. 2014.

[60] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *NeurIPS*, 2017.

[61] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NeurIPS*. 2017.

[62] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *CVPR*, 2017.

[63] Yevhen Kuznietsov, Jorg Stuckler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *CVPR*, July 2017.

[64] Ravi Garg, Vijay Kumar B.G., Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016.

[65] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE Trans. PAMI*, 36(11):2144–2158, 2014.

[66] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, pages 239–248, 2016.

[67] Clement Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

[68] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[69] Venkataraman Santhanam, Vlad I. Morariu, and Larry S. Davis. Generalized deep image to image regression. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[70] Matan Sela, Elad Richardson, and Ron Kimmel. Unrestricted facial geometry reconstruction using image-to-image translation. In *ICCV*, 2017.

[71] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image and Vision Computing*, 28(5):807 – 813, 2010. Best of Automatic Face and Gesture Recognition 2008.

[72] Koutilya PNVR, Bharat Singh, Pallabi Ghosh, Behjat Siddiquie, and David Jacobs. Ld-znet: A latent diffusion approach for text-based image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4157–4168, October 2023.

[73] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[74] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014.

[75] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[76] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

[77] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2211.09800*, 2022.

[78] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *European Conference on Computer Vision*, pages 108–124. Springer, 2016.

[79] Chenxi Liu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, and Alan Yuille. Recurrent multimodal interaction for referring image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1271–1280, 2017.

[80] Hengcan Shi, Hongliang Li, Fanman Meng, and Qingbo Wu. Key-word-aware network for referring expression image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 38–54, 2018.

[81] Ruiyu Li, Kaican Li, Yi-Chun Kuo, Michelle Shu, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Referring image segmentation via recurrent refinement networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2018.

[82] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10502–10511, 2019.

[83] Edgar Margffoy-Tuay, Juan C Pérez, Emilio Botero, and Pablo Arbeláez. Dynamic multimodal instance segmentation guided by natural language queries. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 630–645, 2018.

[84] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1307–1315, 2018.

[85] Zhaoqing Wang, Yu Lu, Qiang Li, Xunqiang Tao, Yandong Guo, Mingming Gong, and Tongliang Liu. Cris: Clip-driven referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11686–11695, 2022.

[86] Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip HS Torr. Lavt: Language-aware vision transformer for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18155–18165, 2022.

[87] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7086–7096, 2022.

[88] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[89] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1780–1790, 2021.

[90] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Harold Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. In *Advances in Neural Information Processing Systems*, 2022.

[91] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023.

[92] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once, 2023.

[93] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[94] Ming Tao, Hao Tang, Fei Wu, Xiao-Yuan Jing, Bing-Kun Bao, and Changsheng Xu. Df-gan: A simple and effective baseline for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16515–16525, 2022.

[95] Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. Cross-modal contrastive learning for text-to-image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 833–842, 2021.

[96] Hui Ye, Xiulong Yang, Martin Takac, Rajshekhar Sunderraman, and Shihao Ji. Improving text-to-image synthesis using contrastive learning. *The 32nd British Machine Vision Conference (BMVC)*, 2021.

[97] Yufan Zhou, Ruiyi Zhang, Changyou Chen, Chunyuan Li, Chris Tensmeyer, Tong Yu, Jiuxiang Gu, Jinhui Xu, and Tong Sun. Towards language-free training for text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17907–17917, 2022.

[98] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[99] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. Cogview: Mastering text-to-image generation via transformers. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 19822–19835. Curran Associates, Inc., 2021.

[100] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. *arXiv preprint arXiv:2203.13131*, 2022.

[101] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[102] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

[103] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

[104] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[105] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[106] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10696–10706, 2022.

[107] Zhicong Tang, Shuyang Gu, Jianmin Bao, Dong Chen, and Fang Wen. Improved vector quantized diffusion models. *arXiv preprint arXiv:2205.16007*, 2022.

[108] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[109] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[110] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022.

[111] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11461–11471, June 2022.

[112] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. *arXiv preprint arXiv:2212.05034*, 2022.

[113] Jianbo Chen, Yelong Shen, Jianfeng Gao, Jingjing Liu, and Xiaodong Liu. Language-based image editing with recurrent attentive models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8721–8729, 2018.

[114] Andrey Voynov, Stanislav Morozov, and Artem Babenko. Object segmentation without labels with large-scale generative models. In *International Conference on Machine Learning*, pages 10596–10606. PMLR, 2021.

[115] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Finding an unsupervised image segmenter in each of your deep generative models. *arXiv preprint arXiv:2105.08127*, 2021.

[116] Daniil Pakhomov, Sanchit Hira, Narayani Wagle, Kemar E. Green, and Nassir Navab. Segmentation in style: Unsupervised semantic image segmentation with stylegan and clip, 2021.

[117] Daiqing Li, Junlin Yang, Karsten Kreis, Antonio Torralba, and Sanja Fidler. Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[118] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020.

[119] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[120] Li Fei-Fei, Robert Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

[121] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[122] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[123] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[124] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[125] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. *Advances in neural information processing systems*, 29, 2016.

[126] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021.

[127] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

[128] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–II. IEEE, 2004.

[129] Fernando De la Torre and Michael J Black. Robust principal component analysis for computer vision. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pages 362–369. IEEE, 2001.

[130] Chenyun Wu, Zhe Lin, Scott Cohen, Trung Bui, and Subhransu Maji. Phrasecut: Language-based image segmentation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10216–10225, 2020.

[131] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014.

[132] Varun K. Nagaraja, Vlad I. Morariu, and Larry S. Davis. Modeling context between objects for referring expression understanding. In *ECCV*, 2016.

[133] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models, 2022.

[134] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models, 2023.

[135] Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models, 2023.

[136] Xianfan Gu, Chuan Wen, Jiaming Song, and Yang Gao. Seer: Language instructed video prediction with latent diffusion models, 2023.

[137] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023.

[138] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7346–7356, 2023.

[139] Jie An, Songyang Zhang, Harry Yang, Sonal Gupta, Jia-Bin Huang, Jiebo Luo, and Xi Yin. Latent-shift: Latent diffusion with temporal shift for efficient text-to-video generation. *arXiv preprint arXiv:2304.08477*, 2023.

[140] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[141] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023.

[142] Alex Hanson, Koutilya PNVR, Sanjukta Krishnagopal, and Larry Davis. Bidirectional convolutional lstm for the detection of violence in videos. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, September 2018.

[143] Guodong Guo and Alice Lai. A survey on still image based human action recognition. *Pattern Recognition*, 47(10):3343–3361, 2014.

[144] Xiaojiang Peng and Cordelia Schmid. Multi-region two-stream r-cnn for action detection. In *European Conference on Computer Vision*, pages 744–759. Springer, 2016.

[145] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2678–2687, 2016.

[146] Swathikiran Sudhakaran and Oswald Lanz. Learning to detect violent videos using convolutional long short-term memory. In *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*, pages 1–6. IEEE, 2017.

[147] Tao Zhang, Wenjing Jia, Xiangjian He, and Jie Yang. Discriminative dictionary learning with motion weber local descriptor for violence detection. *IEEE Trans. Cir. and Sys. for Video Technol.*, 27(3):696–709, March 2017.

[148] Zhihong Dong, Jie Qin, and Yunhong Wang. Multi-stream deep networks for person to person violence detection in videos. In Tieniu Tan, Xuelong Li, Xilin Chen, Jie Zhou, Jian Yang, and Hong Cheng, editors, *Pattern Recognition*, pages 517–531, Singapore, 2016. Springer Singapore.

[149] Piotr Tadeusz Bilinski and François Brémond. Human violence recognition and detection in surveillance videos. *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 30–36, 2016.

[150] Yuan Gao, Hong Liu, Xiaohu Sun, Can Wang, and Yi Liu. Violence detection using oriented violent flows. *Image Vision Comput.*, 48(C):37–41, April 2016.

[151] Yu Zhang, William Chan, and Navdeep Jaitly. Very deep convolutional networks for end-to-end speech recognition. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4845–4849, 2017.

[152] Yan Huang, Wei Wang, and Liang Wang. Bidirectional recurrent convolutional networks for multi-frame super-resolution. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 235–243. Curran Associates, Inc., 2015.

[153] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao. A multi-stream bidirectional recurrent neural network for fine-grained action detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1961–1970, June 2016.

[154] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *In International Conference on Learning Representations*, 2015.

[155] Enrique Bermejo Nievas, Oscar Deniz Suarez, Gloria Bueno García, and Rahul Sukthankar. Violence detection in video using computer vision techniques. In *International conference on Computer analysis of images and patterns*, pages 332–339. Springer, 2011.

[156] T. Hassner, Y. Itcher, and O. Kliper-Gross. Violent flows: Real-time detection of violent crowd behavior. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–6, June 2012.

[157] J. Nam, M. Alghoniemy, and A. H. Tewfik. Audio-visual content-based violent scene characterization. In *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269)*, volume 1, pages 353–357 vol.1, Oct 1998.

[158] Theodoros Giannakopoulos, Dimitrios Kosmopoulos, Andreas Aristidou, and Sergios Theodoridis. Violence content classification using audio features. In *Hellenic Conference on Artificial Intelligence*, pages 502–507. Springer, 2006.

[159] Jian Lin and Weiqiang Wang. Weakly-supervised violence detection in movies with audio and video based co-training. In *Pacific-Rim Conference on Multimedia*, pages 930–935. Springer, 2009.

[160] Hossein Mousavi, Sadegh Mohammadi, Alessandro Perina, Ryad Chellali, and Vittorio Murino. Analyzing tracklets for the detection of abnormal crowd behavior. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 148–155. IEEE, 2015.

[161] Roberto Olmos, Siham Tabik, and Francisco Herrera. Automatic handgun detection alarm in videos using deep learning. *Neurocomputing*, 275:66–72, 2018.

[162] Oscar Deniz, Ismael Serrano, Gloria Bueno, and Tae-Kyun Kim. Fast violence detection in video. In *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, volume 2, pages 478–485. IEEE, 2014.

[163] Ismael Serrano Gracia, Oscar Deniz Suarez, Gloria Bueno Garcia, and Tae-Kyun Kim. Fast fight detection. *PLoS one*, 10(4):e0120448, 2015.

[164] Datong Chen, Howard Wactlar, Ming-yu Chen, Can Gao, Ashok Bharucha, and Alex Hauptmann. Recognition of aggressive human behavior using binary local motion descriptors. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 5238–5241. IEEE, 2008.

[165] Long Xu, Chen Gong, Jie Yang, Qiang Wu, and Lixiu Yao. Violent video detection based on mosift feature and sparse coding. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 3538–3542. IEEE, 2014.

[166] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2017.

[167] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.

[168] Jefferson Ryan Medel and Andreas E. Savakis. Anomaly detection in video using predictive convolutional long short-term memory networks. *CoRR*, abs/1612.00390, 2016.

[169] Viorica Patraucean, Ankur Handa, and Roberto Cipolla. Spatio-temporal video autoencoder with differentiable memory. *arXiv preprint arXiv:1511.06309*, 2015.

[170] Mike Schuster and Kuldip K Paliwal. Bidirectional Recurrent Neural Networks. *IEEE Transactions on Signal Processing*, 45(11), 1997.

[171] Alex Graves, Navdeep Jaitly, and Abdel rahman Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *In IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU*, 2013.

[172] Zhiyong Cui, Ruimin Ke, and Yinhai Wang. Deep bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction. *CoRR*, abs/1801.02143, 2018.

[173] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm networks. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 4, pages 2047–2052 vol. 4, July 2005.

[174] Y. Huang, W. Wang, and L. Wang. Video super-resolution via bidirectional recurrent convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):1015–1028, April 2018.

[175] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning video object segmentation with visual memory. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4491–4500, 2017.

[176] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 568–576. Curran Associates, Inc., 2014.

[177] J. Deng, W. Dong, R. Socher, L. J. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009.

[178] Long Xu, Chen Gong, Jie Yang, Qiang Wu, and Lixiu Yao. Violent video detection based on mosift feature and sparse coding. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3538–3542, 2014.

[179] Oscar Déniz-Suárez, Ismael Serrano, Gloria Bueno García, and Tae-Kyun Kim. Fast violence detection in video. *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, 2:478–485, 2014.

[180] Sadegh Mohammadi, Hamed Kiani, Alessandro Perina, and Vittorio Murino. Violence detection in crowded scenes using substantial derivative. In *2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, aug 2015.

[181] Swathikiran Sudhakaran. Personal communication.