A REPORT

ON

<u>FACE RECOGNITION MODEL</u>

BY

Vinamra Mahajan 2021A7PS2695P

AT

NORTHCORP SOFTWARE, GURUGRAM

A Practice School-1 Station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

(June, 2023)

A REPORT

ON

FACE RECOGNITION MODEL

BY

Vinamra Mahajan 2021A7PS2695P

Prepared in partial fulfillment of the

Practice School-1 Course Nos.

BITS C221/BITS C231/BITS C241

AT

NORTHCORP SOFTWARE, GURUGRAM

A Practice School-1 Station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

(June, 2023)

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude and appreciation to my colleagues at Northcorp Software for their unwavering support and valuable contributions throughout the ongoing development of the Face Recognition Model. Their insightful discussions, constructive feedback, and willingness to share their knowledge and expertise have significantly enriched the quality of my work.

I would also like to extend my gratitude to the management and leadership team at Northcorp Software for providing me with the opportunity to work on this project. Their support, guidance, and belief in my abilities have been crucial in driving the advancement of the facial recognition model.

I would also like to acknowledge the technical resources and infrastructure provided by the organization. These resources have been indispensable in facilitating the development and experimentation required for the model.

Lastly, I would like to extend my heartfelt appreciation to the faculty in charge, for their unwavering support and consistent guidance throughout the development of the Model. Their moral support and regular sessions have been invaluable in keeping us motivated and focused on the project. Their commitment to creating a supportive learning environment has fostered an atmosphere of growth and exploration, allowing me to delve into the intricacies of facial recognition technology with confidence.

I am sincerely grateful to each and every colleague at Northcorp Software for their invaluable support and commitment to this project. Your contributions have been integral to its progress, and I am honored to work alongside such a talented and dedicated team.

I would like to express my sincere gratitude to the University of Michigan for offering comprehensive courses in machine learning, computer vision, and data science. The valuable insights and learnings gained from these courses have played a crucial role in shaping the development and implementation of the Facial Recognition Model, enabling me to apply cutting-edge techniques and make informed decisions throughout the project.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI

(RAJASTHAN)

Practice School Division

Station: Northcorp Software Centre: Gurugram

Duration: 3 Weeks Date of Start: 02/06/2023

Date of Submission: 19/07/23

Title of Project: Face Recognition Model

2021A7PS2695P Vinamra Mahajan Computer Science

Names and Designation of the experts:

Rohit Pandey Business Analyst

Rishi Shukla Project Manager

Name of the PS Faculty: Sunny Singh

Keywords: OpenCV, Machine Learning, Deep Neural Networks, Computer Vision, Face Extraction.

Project Areas: The project areas of face recognition includes face detection, feature extraction, face matching, deep neural networks, classification algorithms, data preprocessing, and model evaluation. These components collectively contributed to accurate face identification, precise feature extraction, reliable matching, improved accuracy through deep learning techniques, enhanced performance through classification algorithms, efficient data handling using pandas, and thorough model evaluation for reliability and robustness.

Abstract:

This project focuses on the development of a face recognition model using OpenCV, TensorFlow, Scikit-learn, and pandas. The model will comprise of key components such as face detection, feature extraction, face matching, deep neural networks, classification algorithms, data preprocessing, and model evaluation.

Face detection accurately identifies and localizes human faces in images or videos. Feature extraction extracts unique facial landmarks, enabling precise recognition. Face matching mechanisms compare extracted features with a database, ensuring reliable identification.

Deep neural networks powered by TensorFlow enhance accuracy by efficiently extracting and recognizing facial features. Classification algorithms from Scikit-learn improve performance and categorization processes. Data preprocessing using pandas streamlines data handling and organization, contributing to the model's efficiency.

Model evaluation methodologies assess the model's performance, ensuring its reliability and robustness. The project benefits from knowledge acquired through University of Michigan courses in machine learning, computer vision, and data science.

Overall, this facial recognition model demonstrates accurate face identification, precise feature extraction, reliable face matching, and enhanced performance through deep learning and classification algorithms. The integration of OpenCV, TensorFlow, Scikit-learn, and pandas enables the development of a reliable and efficient facial recognition system with potential applications in security, access control, and personalized experiences.

Signature of Student: Signature of PS Faculty:

Date: 19/07/2023 Date:

TABLE OF CONTENTS

INTRODUCTION

Face recognition technology has rapidly emerged as a groundbreaking field with a wide range of applications spanning security, surveillance, authentication, and personalization. The ability of machines to accurately identify and analyze human faces has opened up new possibilities for improving efficiency, enhancing security measures, and delivering personalized user experiences. In this report, we delve into the development and implementation of a Facial Recognition Model, aiming to explore its potential, challenges, and implications.

The primary objective of this report is to provide an overview of the ongoing work on the Facial Recognition Model and shed light on its significance in today's technological landscape. By leveraging advancements in computer vision, pattern recognition, and machine learning, our model seeks to analyze facial features, extract relevant information, and match it against a database of known individuals.

The Facial Recognition Model holds immense potential in various domains. In the realm of security, it can enhance surveillance systems by identifying potential threats or unauthorized individuals in real-time. Furthermore, facial recognition can streamline authentication processes, enabling secure access to devices, applications, or physical spaces. Additionally, this technology has the capacity to enable personalized experiences in various industries, such as marketing, entertainment, and healthcare.

However, the development of an accurate and reliable Facial Recognition Model poses certain challenges. Issues such as occlusions, variations in lighting conditions, pose, and expression can impact the model's performance. Furthermore, ethical considerations surrounding privacy and potential biases need to be addressed to ensure responsible implementation of facial recognition technology.

This report will detail the methodology employed in developing the Facial Recognition Model, including data collection, preprocessing techniques, feature extraction, and the underlying model architecture. It will also discuss the training process, evaluation metrics, and the analysis of results obtained thus far.

Overall, this report aims to provide a comprehensive overview of the ongoing work on the Facial Recognition Model, exploring its potential benefits, challenges, and implications. By gaining insights into the current state of the model, we can pave the way for future enhancements and responsible deployment of facial recognition technology in various domains.

Face Detection using TensorFlow

Face detection is a crucial initial step in building a face recognition model, as it involves locating and identifying the presence of faces within an image or a video stream. TensorFlow, a popular deep learning framework, offers various pre-trained models that can be employed for face detection tasks, such as the Single Shot Multibox Detector (SSD) and You Only Look Once (YOLO).

Single Shot Multibox Detector (SSD):
SSD is an object detection algorithm capable of detecting multiple objects, including faces, in a single pass through the neural network. It combines high detection accuracy with real-time processing speeds, making it suitable for face detection in various applications. The SSD model consists of a base convolutional network, which extracts image features, followed by additional convolutional layers responsible for detecting objects' bounding boxes and class labels.

During training, the SSD model is presented with annotated images that contain labeled bounding boxes around the faces. The model learns to adjust its parameters to accurately predict the locations of faces in new, unseen images. The loss function used in training penalizes incorrect predictions and rewards accurate bounding box detections.

After training, the SSD model can be integrated into the TensorFlow pipeline for face detection. Given an input image, the model processes it through its layers, predicting the bounding boxes that encapsulate the facial features. These bounding boxes are accompanied by confidence scores, representing the model's confidence in the presence of a face within each box.

You Only Look Once (YOLO):
YOLO is another popular object detection algorithm that provides real-time detection capabilities. Unlike SSD, YOLO follows a single-step detection approach, where the entire image is analyzed at once to generate bounding box predictions and class probabilities. This results in faster processing speeds and reduced computational overhead.

The YOLO model is trained on a similar dataset with annotated face bounding boxes, learning to predict bounding box coordinates and class probabilities directly from the image. During training, the loss function ensures that the model accurately predicts the faces' positions and classifies them correctly.

In TensorFlow, the trained YOLO model can be integrated into the face recognition pipeline to perform real-time face detection. The model processes the input image as a whole and outputs bounding boxes containing faces, along with their corresponding confidence scores.

TensorFlow Integration:
Regardless of whether SSD or YOLO is chosen for face detection, TensorFlow simplifies the integration of these models into the face recognition pipeline. TensorFlow provides high-level APIs, such as TensorFlow Object Detection API, that facilitate the use of pre-trained models and streamline the deployment process.

With TensorFlow's ease of use and scalability, face detection becomes a seamless part of the overall face recognition model. The face detection model's efficient integration ensures real-time performance and accurate face localizations, laying the groundwork for subsequent face recognition tasks to identify and verify individuals based on their unique facial features. By leveraging these powerful deep learning models, TensorFlow significantly contributes to the success of the face recognition system, making it a robust and practical solution for various real-world applications.

Face Recognition with TensorFlow

Face recognition using TensorFlow involves leveraging deep learning techniques, particularly Convolutional Neural Networks (CNNs), to extract essential features from detected faces and create compact representations known as embeddings. These embeddings encode the unique facial characteristics of individuals, enabling the model to identify and verify individuals based on these learned features.

Convolutional Neural Networks (CNNs):
CNNs are a class of deep learning models specifically designed for image processing tasks. They consist of multiple layers of interconnected neurons, including convolutional layers, pooling layers, and fully connected layers. CNNs are adept at learning hierarchical patterns from images, making them suitable for extracting meaningful features from faces.

Custom CNN Architectures:
TensorFlow provides a flexible platform to build and train custom CNN architectures tailored for face recognition. Researchers and developers can design CNN models with various configurations of layers, depths, and kernel sizes, allowing for experimentation and optimization.

Dataset Preparation:
Before training the face recognition model, a dataset of labeled face images is required. This dataset should consist of images of individuals, each labeled with their respective identity or unique identifier. The larger and more diverse the dataset, the better the model's ability to generalize and recognize faces from different demographics and variations.

Encoding Facial Characteristics - Embeddings:
The core objective of the face recognition model is to encode the facial characteristics of each individual into compact feature vectors called embeddings. These embeddings should possess the property of being distinct for different individuals and exhibit similarities for images of the same person.

Training the Face Recognition Model:
During the training phase, the custom CNN architecture is presented with batches of labeled face images from the dataset. The CNN processes these images through its layers, gradually learning to identify discriminative facial features that differentiate one person from another.

The training process involves optimizing the model's parameters using an appropriate loss function, such as triplet loss or contrastive loss. These loss functions enforce the embedding vectors to have small distances for images of the same person and large distances for images of different individuals, thereby promoting better face recognition performance.

Fine-Tuning and Hyperparameter Optimization:

To achieve optimal performance, hyperparameters, such as learning rate, batch size, and regularization parameters, need to be fine-tuned during training. Additionally, transfer learning can be employed by initializing the model with weights from pre-trained CNN models, such as VGGFace or FaceNet, which have been trained on large-scale face recognition datasets.

Embedding Space:
Once the model is trained, it projects each face image into the embedding space, where similar faces cluster together based on their learned features. During recognition or verification tasks, the model compares the embeddings of the query face with those in the database to find the closest matches, allowing it to identify the person or verify their identity.

Utilizing a Face Database

Utilizing a diverse and comprehensive face database is critical for the success of a face recognition model. A face database serves as the primary source of training and evaluating the model, allowing it to learn and generalize patterns from a wide range of individuals. TensorFlow facilitates the integration of such databases into the training pipeline, enabling the model to recognize and identify faces it has not encountered during training. Let's delve into the importance and details of using face databases for training a face recognition model:

Importance of a Diverse Face Database:
A diverse face database is essential for training a robust face recognition model capable of handling variations in appearance, lighting conditions, poses, and facial expressions. The diversity of the dataset ensures that the model is exposed to faces from different ethnicities, genders, age groups, and cultural backgrounds. By including a wide range of individuals in the dataset, the model becomes more inclusive and unbiased, making it applicable to various real-world scenarios.

Publicly Available Face Databases:
There are several publicly available face databases, widely used by the research community for face recognition tasks. Two prominent examples are:

Labeled Faces in the Wild (LFW): LFW is a well-known face database that contains images collected from the internet. It includes images of individuals in uncontrolled environments, exhibiting variations in illumination, background, and image quality. Each image in LFW is labeled with the identity of the person it portrays, allowing the model to learn to associate specific facial features with each person.

VGGFace: VGGFace is a large-scale face database created by the Visual Geometry Group (VGG) at the University of Oxford. It consists of a vast collection of face images, covering a wide variety of identities from around the world. VGGFace enables the model to learn more generalized features, as it includes faces with different poses, expressions, and occlusions.

Data Labeling:
The face database comes with pre-labeled identities, where each image is associated with the corresponding person's identity. This labeled data is crucial for supervised learning, enabling the model to

learn to recognize specific individuals during training. The labels act as ground truth, guiding the model to adjust its parameters to make accurate predictions during face recognition tasks.

Data Preprocessing:
Before integrating the face database into TensorFlow, data preprocessing is essential to ensure consistent and standardized input for the model. Preprocessing steps may involve resizing all images to a uniform size, normalizing pixel values to a common scale, and potentially performing data augmentation to increase dataset diversity.

Training and Generalization:
During training, the model is exposed to the labeled face images from the database. By repeatedly presenting batches of images to the model and adjusting its parameters based on the loss function, the model learns to encode unique facial characteristics into embeddings. The diversity of the face database allows the model to generalize and recognize faces of individuals not seen during training, enabling it to be applied to real-world scenarios effectively.

Evaluation and Model Tuning:
After training, the model's performance is evaluated using a separate test set from the face database. This evaluation helps assess the model's accuracy, precision, recall, and other relevant metrics. Based on the evaluation results, hyperparameters and model architectures can be fine-tuned to optimize the model's performance.

Data Preprocessing and Augmentation

Data preprocessing and augmentation are vital steps in preparing the dataset for training a face recognition model using TensorFlow. These techniques help enhance the model's performance by ensuring consistent input data and increasing the diversity of the dataset, allowing the model to generalize better to unseen variations. Let's delve into the details of data preprocessing and augmentation techniques:

Data Preprocessing:

Resizing:
Resizing the input images to a standard size is a common preprocessing step in deep learning tasks. TensorFlow allows for efficient resizing of images to a specific resolution, ensuring that all images have the same dimensions. This step is crucial as it provides a consistent input size for the model, enabling it to process images efficiently during training and inference.

Normalization:
Normalizing pixel values is essential to scale the image data within a specific range. By normalizing the pixel values, typically to the range of [0, 1] or [-1, 1], the model becomes less sensitive to the scale of the data and can converge faster during training. Common normalization techniques include dividing the pixel values by the maximum pixel value (e.g., 255 for 8-bit images) or using z-score normalization.

Image Enhancements:
Image enhancements can be applied as part of data preprocessing to improve the image quality and increase the model's robustness. These enhancements might include contrast adjustments, histogram equalization, and noise reduction techniques. By enhancing the images, the model can better focus on the relevant facial features, even in challenging lighting conditions or low-quality images.

Data Augmentation:

Data augmentation is a powerful technique used to increase the diversity of the training dataset without collecting new labeled data. By applying various transformations to the existing images, the dataset becomes more representative of the real-world scenarios the model is likely to encounter. TensorFlow provides efficient tools for implementing data augmentation techniques:

Rotation:
Rotating the images by a certain angle introduces variations in facial poses. For example, rotating an image by 90 degrees creates a profile view of the face, which helps the model recognize faces from different angles.

Flipping:
Horizontally flipping the images effectively doubles the dataset, creating mirror images of the faces. This technique is particularly useful for scenarios where facial symmetry is significant.

Random Cropping:
Randomly cropping parts of the images introduces variations in facial expressions and helps the model handle faces of different sizes in the input data.

Brightness and Contrast Adjustments:
Adjusting the brightness and contrast of the images can simulate different lighting conditions, making the model more robust to lighting variations in real-world environments.

Gaussian Noise:
Adding Gaussian noise to the images mimics noise present in real-world images and helps the model generalize better to noisy inputs.

Affine Transformations:
Affine transformations, such as scaling and shearing, can be applied to the images, introducing additional variations in facial poses and expressions.

By incorporating data preprocessing and augmentation techniques in TensorFlow, the face recognition model becomes more versatile and robust, capable of handling various real-world scenarios and improving its generalization performance. These techniques effectively increase the diversity of the training dataset, allowing the model to learn more representative features and enhancing its ability to recognize faces accurately and reliably.

Model Evaluation and Accuracy Metrics

Model evaluation and accuracy metrics are crucial aspects of assessing the performance and effectiveness of a face recognition model trained using TensorFlow. Evaluating the model on a separate test set is essential to ensure that it can generalize well to unseen data and accurately recognize individuals. TensorFlow offers a variety of accuracy metrics to measure the model's performance, allowing developers to fine-tune the model and optimize its accuracy and generalization capabilities. Let's explore these concepts in detail:

Test Set:
To evaluate the face recognition model, a separate test set is created from the face database that was used during training. This test set consists of images that the model has not seen during training, making it an unbiased representation of the model's ability to recognize new faces accurately.

Accuracy:
Accuracy is a fundamental metric that measures the percentage of correctly predicted faces over the total number of faces in the test set. It provides an overall assessment of the model's performance. However, accuracy alone may not be sufficient when dealing with imbalanced datasets, where the number of samples for each identity is unequal.

Precision:
Precision is a metric that quantifies the number of true positive face identifications over the total number of positive predictions made by the model. In the context of face recognition, precision denotes the proportion of correctly identified faces out of all the faces the model has predicted as belonging to a particular individual. A high precision value indicates that the model has a low rate of false positives.

Recall (Sensitivity or True Positive Rate):
Recall measures the number of true positive face identifications divided by the total number of actual positive faces in the test set. It represents the ability of the model to correctly identify all instances of a particular individual. A high recall value indicates that the model can effectively identify the majority of faces belonging to a specific individual.

F1-score:
The F1-score is the harmonic mean of precision and recall, providing a balanced measure of the model's accuracy. It is particularly useful when the dataset is imbalanced. The F1-score ranges from 0 to 1, where 1 indicates perfect precision and recall. A higher F1-score suggests a more robust face recognition model.

Confusion Matrix:
A confusion matrix is a tabular representation that summarizes the model's predictions and their actual outcomes. It provides insights into true positives, true negatives, false positives, and false negatives. From the confusion matrix, accuracy, precision, recall, and other metrics can be derived.

Fine-tuning the Model:
After evaluating the model's performance using the above metrics, developers can fine-tune the model to

improve its accuracy and generalization capabilities. Fine-tuning may involve adjusting hyperparameters, modifying the model architecture, or increasing the size and diversity of the training dataset.

Real-Time Application and Deployment

Real-time application and deployment of a face recognition model using TensorFlow is crucial for its practical usability in various real-world scenarios. TensorFlow's compatibility with diverse hardware and its optimization for deployment enable the model to efficiently process incoming video streams in real-time. Let's delve into the details of how TensorFlow facilitates real-time application and deployment:

Compatibility with Various Hardware:
TensorFlow is designed to be hardware agnostic, meaning it can seamlessly run on different hardware platforms, including CPUs, GPUs (Graphics Processing Units), and TPUs (Tensor Processing Units). This flexibility allows developers to choose the most suitable hardware for their specific real-time face recognition application.

GPU and TPU Acceleration:
Specialized hardware accelerators like GPUs and TPUs significantly speed up the computational tasks performed by the face recognition model. GPUs excel at parallel processing, which is well-suited for deep learning operations, making them ideal for speeding up training and inference tasks. TPUs, developed by Google, are specifically designed for handling deep learning workloads and offer even more significant performance gains, particularly in large-scale deployments.

Optimization for Real-Time Inference:
TensorFlow provides optimizations for efficient real-time inference, enabling the face recognition model to process video streams with low latency. These optimizations include model quantization (converting model weights to lower precision), model pruning (removing less important weights), and model compression techniques. By reducing the model's size and computational complexity, real-time inference becomes more feasible without sacrificing accuracy significantly.

TensorFlow Serving:
TensorFlow Serving is a dedicated component that allows developers to deploy trained TensorFlow models easily. It provides a high-performance serving API that handles multiple requests in parallel, making it well-suited for real-time applications. TensorFlow Serving also supports versioning, allowing for seamless updates and rollback of deployed models.

TensorFlow Lite:
TensorFlow Lite is a lightweight version of TensorFlow specifically designed for mobile and edge devices. It enables developers to deploy face recognition models on resource-constrained devices, such as smartphones and IoT (Internet of Things) devices. TensorFlow Lite further facilitates real-time applications by ensuring efficient use of computational resources.

Real-Time Face Recognition Applications:
The efficient deployment of face recognition models using TensorFlow opens up a wide range of real-time

applications, including:

Security Systems: Real-time face recognition can be utilized for access control, surveillance, and identity verification in secure facilities.
Personalization: Applications can use real-time face recognition to personalize user experiences, such as personalized content recommendations or user-specific settings.
User Authentication: Real-time face recognition can serve as a biometric authentication method for unlocking devices or accessing sensitive data.

CONCLUSION

In conclusion, TensorFlow emerges as a powerful and versatile tool for building, training, and deploying face recognition models with exceptional performance and real-time capabilities. By harnessing deep learning techniques, such as Convolutional Neural Networks (CNNs), TensorFlow excels at extracting and encoding relevant facial features, which are crucial for accurate face recognition.

The success of a face recognition model is heavily dependent on the availability of a diverse and comprehensive face database. TensorFlow efficiently handles data preprocessing and augmentation, ensuring that the model is exposed to various facial expressions, poses, and lighting conditions. With publicly available face databases like Labeled Faces in the Wild (LFW) and VGGFace, the model gains the ability to generalize effectively and recognize individuals it has not encountered during training.

During model evaluation, TensorFlow provides a suite of accuracy metrics, including accuracy, precision, recall, and F1-score, which enable a comprehensive assessment of the model's performance. By fine-tuning the model based on evaluation results, developers can optimize its accuracy and generalization capabilities, ensuring its effectiveness in real-world applications.

The compatibility of TensorFlow with various hardware, including GPUs and TPUs, facilitates efficient processing of video streams in real-time. This compatibility, combined with TensorFlow Serving and TensorFlow Lite, enables the seamless deployment of face recognition models on servers, edge devices, and mobile platforms. Real-time applications, such as security systems, access control, and personalized user experiences, benefit from TensorFlow's ability to handle parallel processing, low-latency inference, and efficient use of computational resources.

The utilization of TensorFlow in real-time face recognition empowers various industries, offering improved security, streamlined access control, and personalized user interactions. From enhancing surveillance systems to unlocking the potential for personalized content delivery, TensorFlow's capabilities provide a robust and practical solution to the challenges posed by face recognition in modern technological environments.

REFERENCES

1. Machine Learning Course by University of Michigan: https://www.coursera.org/learn/python-machine-learning/
2. Data Science Basics by University of Michigan: https://www.coursera.org/learn/python-data-analysis/
3. OpenCV Python tutorials: https://youtube.com/playlist?list=PLS1QulWo1RIa7D1O6skqDQ-JZ1GGHKK-K
4. Python libraries documents: https://docs.python.org/3/library/

GLOSSARY

Machine Learning: Machine learning is a subfield of artificial intelligence that focuses on the development of algorithms and models that enable computers to learn from and make predictions or decisions based on data. In the Facial Recognition Model, machine learning techniques are employed for tasks such as face detection, feature extraction, and face matching.

Computer Vision: Computer vision is a field of study that deals with how computers can gain a high-level understanding from digital images or videos. In the context of the Facial Recognition Model, computer vision techniques are used for tasks such as detecting faces, extracting facial features, and matching faces to a database.

Data Science: Data science is an interdisciplinary field that combines statistics, mathematics, and programming to extract insights and knowledge from data. In the context of the Facial Recognition Model, data science principles and techniques are applied for tasks such as data preprocessing, feature selection, and model evaluation.

Dimensionality Reduction: Dimensionality reduction is a technique used to reduce the number of variables or features in a dataset while preserving as much of the original information as possible. It is often employed to address the curse of dimensionality and improve the efficiency and effectiveness of machine learning models.

Model Evaluation: Model evaluation is the process of assessing the performance and generalization capabilities of a machine learning model. It involves measuring various metrics, such as accuracy, precision, recall, and F1 score, to determine how well the model performs on unseen or test data.

Data Exploration: Data exploration is the process of examining and understanding the characteristics, patterns, and relationships within a dataset. It involves tasks such as visualizing data, computing descriptive statistics, and identifying outliers or anomalies.

Visualization: Visualization refers to the representation of data and information using visual elements such as charts, graphs, and plots. It aids in understanding patterns, trends, and relationships within the data, making complex information more accessible and interpretable.

Deep Neural Networks: Deep neural networks, also known as deep learning models, are a type of artificial neural network with multiple hidden layers. They are capable of learning hierarchical representations of data and have achieved significant success in various domains, including computer vision and natural language processing.