Real-Time Facial Expression Recognition & Response System for Children

Pavan Sai Guntha
Department of Electrical Engineering
University of Washington
Seattle, WA 98105
pguntha@uw.edu

Vidhaya Datta Reddy Metta
Department of Electrical Engineering
University of Washington
Seattle, WA 98105
vmetta@uw.edu

Abstract

This paper presents a novel interactive system designed specifically for children that combines real-time facial expression recognition (FER) with responsive animated character generation. Our proposed framework consists of two main components: an accurate front-end FER model based on a hybrid CNN architecture (MobileNetV1 + Vision Transformer [ViT] [10]) achieving 93.71% accuracy on standard datasets, and a back-end conditional Generative Adversarial Network (cGAN) [**ref_2**] that produces contextually appropriate animated character responses. The system processes camera input through the FER model to identify emotional states (happiness, sadness, anger, fear, neutral, disgust, surprise), which then trigger the corresponding cartoon character animations through the cGAN. FER model is trained on established datasets (RAF-DB [7], CK+ [9], FERPlus [2]), while the cGAN was trained on the FERG-DB [1] dataset containing 2D cartoon character expressions. Our architecture is optimized for edge deployment on the NVIDIA Jetson Orin Nano platform, enabling real-time performance while maintaining high accuracy. This system demonstrates significant potential for applications in child-computer interaction, educational technologies, and therapeutic interventions for social-emotional development.

1 Introduction

Human-computer interaction (HCI) has evolved from traditional keyboard and mouse interfaces to more intuitive and natural interaction paradigms. Among these emerging technologies, affective computing, which focuses on systems that can recognize, interpret, and respond to human emotions, represents a particularly promising frontier, especially for applications involving children.

1.1 Motivation

Children engage more deeply with interactive systems that respond to their emotional states, creating more meaningful and personalized experiences. Traditional interactive systems typically lack the ability to perceive and respond to a child's emotional state, resulting in interactions that can feel rigid and impersonal. By developing technology that can recognize a child's expressions and respond appropriately, we can create more engaging, adaptive, and emotionally intelligent interactive experiences.

Such systems have potential applications in key domains:

- Educational technology that adapts teaching strategies based on a child's emotional engagement and therapeutic tools for children with social communication challenges.

- Entertainment systems and assessment tools for tracking emotional development in early childhood.

1.2 Facial Expression Recognition (FER)

Facial Expression Recognition (FER) involves the automated identification of human emotional states from facial images or video sequences. Traditional approaches relied heavily on hand-crafted features such as Local Binary Patterns (LBP) [11], Histogram of Oriented Gradients (HOG) [3], and geometric facial landmarks extracted using Active Appearance Models (AAM) or Active Shape Models (ASM) [6]. These methods required extensive domain expertise and often struggled with variations in lighting, pose, and individual facial characteristics.

1.3 Conditional Generative Adversarial Networks (cGANs)

Generative Adversarial Networks (GANs) have revolutionized the field of image synthesis through an adversarial training process in which a generator network learns to create realistic images, by competing against a discriminator network that attempts to distinguish a real from a generated content. This minimax game results in increasingly sophisticated image generation capabilities.

2 Methods

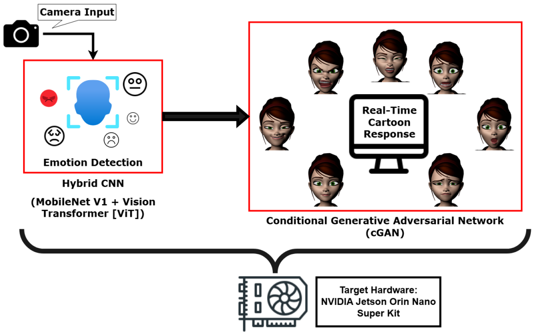

Camera Input
Emotion Detection
Hybrid CNN
(MobileNet V1 + Vision Transformer [ViT])
Real-Time Cartoon Response
Conditional Generative Adversarial Network (cGAN)
Target Hardware: NVIDIA Jetson Orin Nano Super Kit

Figure 1: Complete System Overview

2.1 Front-end Architecture: Facial Expression Recognition

The facial expression recognition system implements an enhanced version of PAtt-Lite [10], a lightweight patch and attention network designed for real-time performance on edge devices. The architecture consists of three main components working in sequence to achieve efficient emotion classification.

2.1.1 Model Architecture

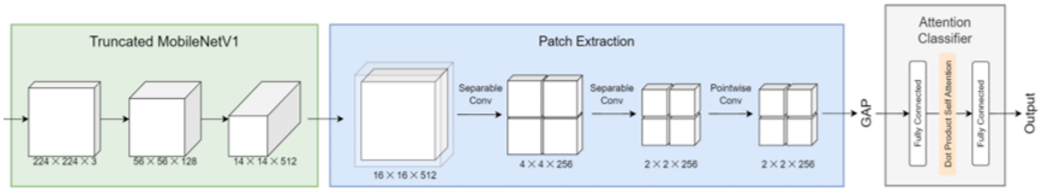


Figure 2: PaTT-Lite: FER Model Architecture [10]

Truncated MobileNetV1 Backbone: The system employs pre-trained MobileNetV1 [5] truncated after block 9's depthwise convolution, resulting 14×14×512 feature maps that preserve lightweight characteristics essential for real-time deployment. MobileNetV1's depthwise separable convolutions achieve 8-9× parameter reduction compared to standard convolutions while maintaining sufficient representational capacity for facial expression patterns.

Enhanced Patch Extraction Module: The patch extraction component processes 14×14×512 feature maps through depthwise separable convolutions, applying padding to achieve 16×16 dimensions before employing two-stage spatial reduction (16×16→8×8→7×7) using 4×4 kernels with stride 2, culminating in adaptive pooling to 2×2×256 patches. Integrated Squeeze-and-Excitation blocks provide channel-wise attention recalibration that enhances focus on emotionally relevant facial features while maintaining computational efficiency.

Multi-Head Attention Classifier: The classification component employs a 4-head self-attention mechanism with 128-dimensional hidden representations to replace traditional fully connected layers, processing globally average-pooled patch features through query, key, and value projections. This attention-based design dynamically weights facial region importance, effectively handling occlusions and pose variations while incorporating residual connections and layer normalization for training stability before final emotion classification across 7 categories through a two-layer feed-forward network.

2.1.2 Dataset Preparation and Normalization

Multi-Dataset Integration: Training combines RAF-DB, CK+, and FER+ datasets with MTCNN face detection [12] with a confidence threshold of 0.8 to filter invalid samples and ensure robust cross-dataset generalization. The final dataset contains 25,306 image samples for training, 5,423 samples for testing and 5,423 samples for validation.

Cross-Dataset Normalization: A critical challenge in multi-dataset training involves handling varying illumination conditions, image qualities, and demographic distributions across datasets. The system Unified normalization parameters (**mean=[0.4873, 0.4873, 0.4873]**, **std=[0.2593, 0.2593, 0.2593]**) derived from the combined dataset distribution ensure consistent pre-processing across varying illumination and demographic conditions.

Enhanced Data Augmentation: A comprehensive augmentation process includes random resized crops (scale 0.8-1.0), horizontal flipping, rotation (±15 degrees), color jittering, affine transformations, perspective distortion, Gaussian blur, and random erasing. Class imbalance is addressed through weighted random sampling based on inverse frequency weighing.

2.1.3 Two-Stage Training Strategy

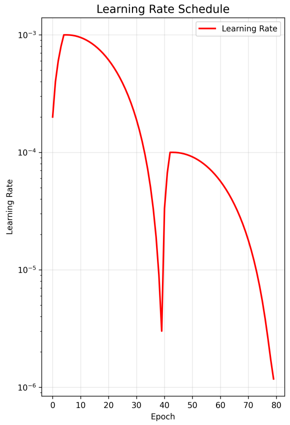


Figure 3: Learning Rate Schedule in 2-Stage Training Approach

Stage 1: Component-Specific Training (Epochs 0-40): The training adopts a two-stage approach over 80 total epochs to optimize convergence and prevent catastrophic forgetting of pre-trained features. Stage 1 focuses exclusively on training the newly introduced patch extraction and attention classifier components while keeping the MobileNetV1 backbone frozen. This strategy allows the new components to adapt to the specific characteristics of facial expression data without disrupting the established feature representations learned from ImageNet pre-training. The stage employs AdamW [8] optimization with warmup scheduling from 1×10^{-4} to 1×10^{-3} over the first 5 epochs, followed by a cosine annealing decay throughout the remaining 35 epochs of Stage 1.

Stage 2: End-to-End Fine-tuning (Epochs 40-80): At Epoch 40, Stage 2 implements the selective unfreezing of the final 60 layers of the MobileNetV1 backbone, enabling end-to-end fine-tuning with a reset learning rate of 1×10^{-4}. This selective unfreezing approach balances adaptation to facial expression patterns while preserving lower-level feature representations. Stage 2 employs cosine annealing from the 1×10^{-4} starting point over the final 40 epochs, with gradient clipping (maximum norm of 1.0) to maintain training stability and label smoothing (α=0.1) to improve generalization performance.

2.2 Rule-Based Emotion Mapping System

The decision system serves as the critical bridge between facial expression recognition and therapeutic response generation, implementing evidence-based emotional regulation strategies through structured sequence mapping. This subsystem transforms detected emotions into temporally organized response patterns that guide the animated character's behavior according to established child psychology principles.

2.3 Dual-Mode Operation

2.3.1 Therapeutic Mode

The therapeutic mode implements a four-stage emotional regulation protocol designed to support children's emotional development: validation → gradual transition → positive reinforcement → stabilization. The system generates response sequences that begin with validation through brief emotion mirroring, followed by gradual transition toward emotional regulation states. Example sequences:

- Anger: anger(20) → neutral(25) → happy(30) → neutral(15)
- Fear: neutral(20) → fear(15) → neutral(25) → happy(25)

2.3.2 Mirror Mode

The mirror mode provides direct emotional reflection, maintaining the detected emotion for extended periods to offer validation and emotional acknowledgment. This mode proves particularly valuable for children, when a simple empathetic response is more appropriate than an active regulation. Example sequences:

- Sadness: sadness(60) → sadness(45)
- Anger: anger(50) → anger(40)

2.4 Back-end Architecture: Animated Character Response Generation

The 256x256 conditional GAN employs a dual-network design optimized for emotion-conditioned facial expression generation while maintaining real-time performance.

Generator Architecture: The generator accepts a 128-dimensional noise vector and emotion label (7 classes), using learned embedding to map labels to 64-dimensional space. A Six-stage transposed convolution upsampling progressively doubles spatial resolution: 4×4×1024 → 8×8×512 → 16×16×256 → 32×32×128 → 64×64×64 → 128×128×32 → 256×256×16. Each stage employs 4×4 kernels with stride-2, batch normalization, and ReLU activation. The architecture uses Xavier initialization [4] and concludes with tanh activation producing [-1, 1] pixel values, plus a 3×3 smoothing convolutional layer to reduce high-frequency artifacts.

Discriminator Architecture: The discriminator processes 256×256×3 RGB images with spatial emotion conditioning through six downsampling stages: 128×128×17 → 64×64×32 → 32×32×64 → 16×16×128 → 8×8×256 → 4×4×512. Input regularization uses Gaussian noise injection (σ=0.1). Each stage implements stride-2 convolutions, batch normalization, LeakyReLU (α=0.2), and dropout

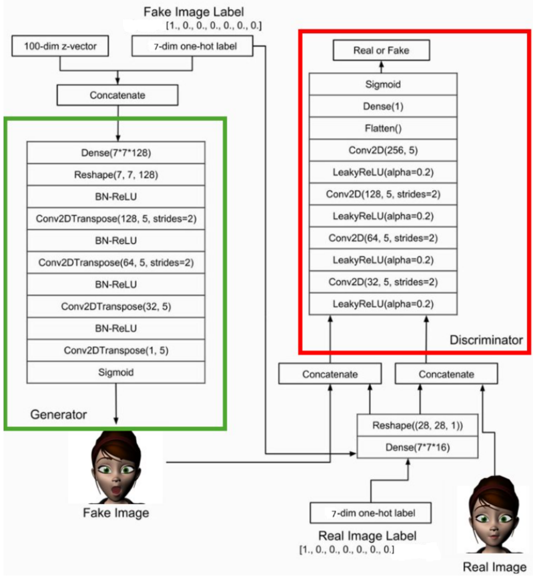


Figure 4: Sample cGAN Architecture: Both generator and discriminator components

(0.25). Emotion labels transform into 128×128×1 feature maps concatenated with image representations. The final pipeline includes minibatch discrimination to prevent mode collapse, followed by a 256-unit dense layer with LeakyReLU and 30% dropout for classification.

Dataset: The FERG-DB [1] dataset is used as the training and testing data for our cGAN model, from which the Mery character is chosen to generate the video-like response.

2.4.1 Training Strategy

The training methodology addresses GAN stability challenges through advanced loss design and optimized scheduling.

Loss Functions: Generator loss combines adversarial binary cross-entropy with diversity regularization measuring intra-class variation through pairwise Euclidean distance calculations and exponential penalties (λ=0.15). Discriminator training uses aggressive label smoothing with real samples receiving 0.8 labels and fake samples 0.2 (instead of 1.0/0.0), creating a forgiving environment for effective generator learning despite high discriminator accuracy.

Training Schedule: Asymmetric update frequencies give the generator two gradient updates per discriminator update, addressing discriminator dominance in high-resolution training. Learning rates reflect this asymmetry: generator (0.0002) versus discriminator (0.0001). Both use Adam optimization with β_1=0.5, β_2=0.999, preventing momentum-based oscillations. Batch size is reduced to 16, while maintaining decent diversity for minibatch discrimination.

2.5 Jetson Orin Nano Deployment

The system employs a multi-threaded pipeline architecture with three dedicated threads handling camera capture, inference processing, and display operations, utilizing frame queues with 2-frame buffers to maintain real-time performance.

2.5.1 TensorRT Acceleration

Both FER and cGAN models are designed to utilize TensorRT [13] 10.3.0 engines with FP16 precision support. Engine conversion employs 512 MB workspace limits and sparse weight optimization flags specifically configured for the Jetson's 8 GB GPU memory constraint.

2.5.2 Optimization Techniques

CUDA Context Management: Global context sharing across threads with push/pop synchronization is implemented to prevent threading conflicts in TensorRT.

Memory Optimization: Conservative memory allocation using regular numpy arrays is leveraged instead of pre-allocated buffers, and fixed batch size is employed for consistent memory usage patterns.

Datatype Precision: FP16 inference support is configured in TensorRT builder flags to reduce memory footprint. PAtt-Lite preprocessing is optimized with vectorized operations and in-place transformations.

Pipeline Efficiency: Frame rate of over 30 FPS is achieved using a queue-based architecture and a conservative cache sizing (5 frames) is employed which balances performance with memory constraints.

3 Results

The enhanced PAtt-Lite model achieved 93.71% test accuracy on the combined multi-dataset evaluation. Per-class performance showed exceptional recognition rates for Anger (97.5%), Disgust (99.6%), and Fear (99.7%), with F1-scores ranging from 0.847 to 0.993.

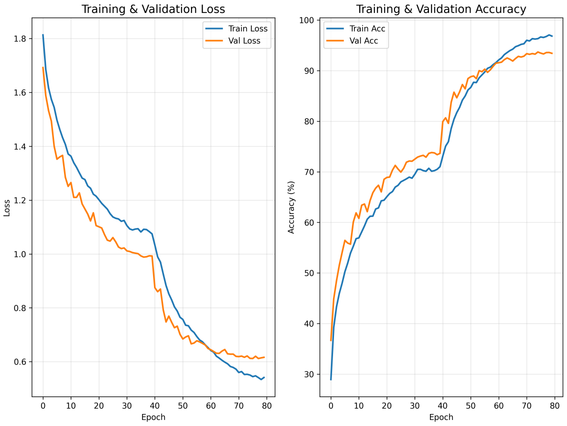


Figure 5: (a) Training and Validation Loss (b) Training and Validation Accuracy

The primary classification challenges occurred between emotionally similar states: ~67 Sadness samples misclassified as Neutral and ~65 Neutral samples as Sadness, reflecting the inherent difficulty in distinguishing subtle emotional transitions. Validation accuracy tracked training performance at ~94%, indicating robust generalization without overfitting.

The cGAN model completed training after 230 epochs with balanced performance (generator loss ~0.77, discriminator loss ~1.36). Diversity analysis across 7 emotion classes ranged from 0.0223 (neutral) to 0.0387 (fear), successfully preventing mode collapse. The system generated consistent emotion-specific outputs with smooth transition animations between emotional states.

Real-time deployment achieved ~30 FPS on NVIDIA Jetson Orin Nano with TensorRT optimization. The generator maintained stable output values between [-1.000, 0.976] and produced high-quality 256×256 facial expressions suitable for pediatric therapeutic applications.

4 Conclusion

This work successfully developed an end-to-end interactive system achieving 93.71% test accuracy for facial expression recognition and ~30 FPS real-time animated character generation across seven emotion classes. The system integrates an enhanced PAtt-Lite architecture with a conditional GAN, optimized for NVIDIA Jetson Orin Nano deployment using TensorRT acceleration. This demon-

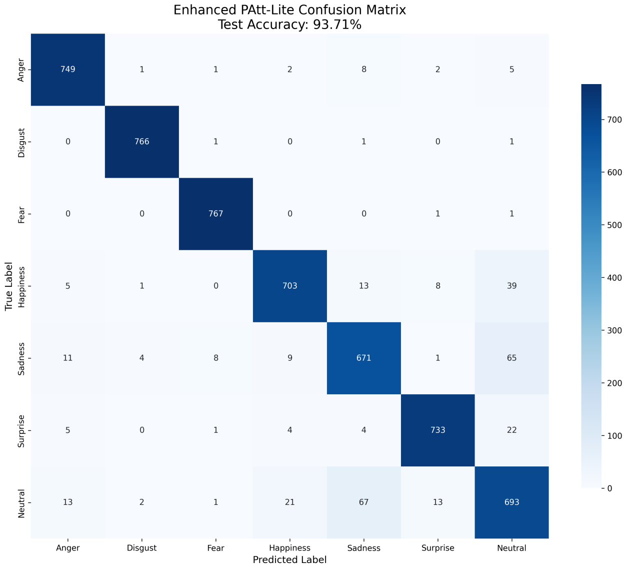


Figure 6: Confusion Matrix illustrating the performance across all the 7 emotion classes

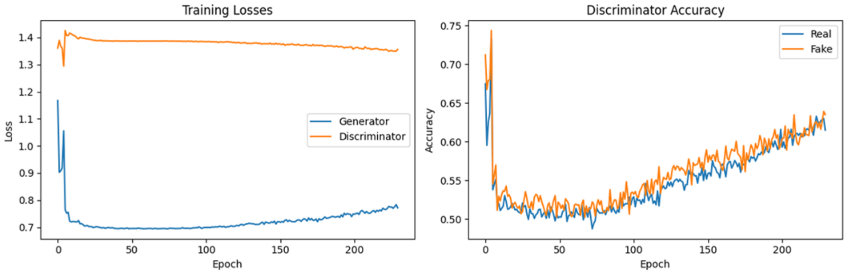
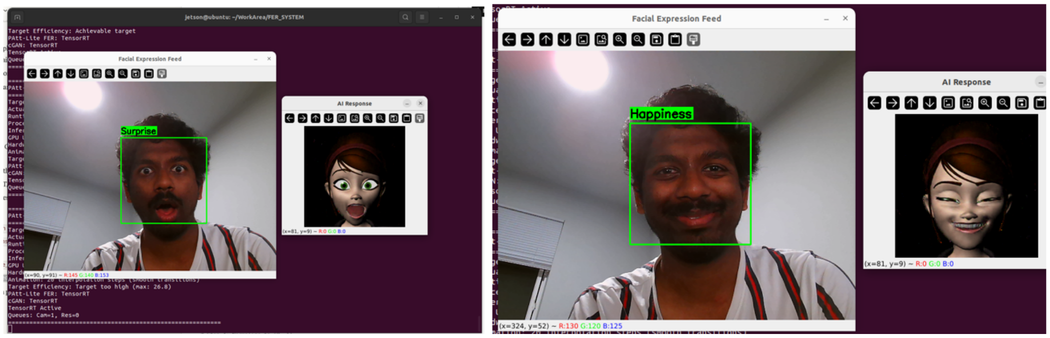


Figure 7: (a) Generator & Discriminator Training Losses (b) Discriminator Accuracy in Real vs Fake Classification


jetson@ubuntu: ~/WorkArea/FER_SYSTEM
Facial Expression Feed
Surprise
(x=90, y=91) ~ R:145 G:140 B:153
AI Response
(x=81, y=9) ~ R:0 G:0 B:0
Facial Expression Feed
Happiness
(x=324, y=52) ~ R:130 G:120 B:125
AI Response
(x=81, y=9) ~ R:0 G:0 B:0

```
Target Efficiency: Achievable target
PAtt-Lite FER: TensorRT
cGAN: TensorRT
Target Efficiency: Target too high (max: 26.8)
PAtt-Lite FER: TensorRT
cGAN: TensorRT
TensorRT Active
Queues: Cam=1, Res=0
============================================================
```

Figure 8: Inference Outcome on NVIDIA Jetson Orin Nano

strates significant potential for transforming child-computer interaction in educational technology, therapeutic interventions, and emotional development assessment through empathetic and responsive digital experiences.

You can find the project code on GitHub: `https://github.com/pavan24sai/fer_cgan_system`.

References

[1] Deepali Aneja et al. "Modeling Stylized Character Expressions via Deep Learning". In: *Asian Conference on Computer Vision*. Springer. 2016, pp. 136–153.

[2] Emad Barsoum et al. "Training Deep Networks for Facial Expression Recognition with Crowd-Sourced Label Distribution". In: *ACM International Conference on Multimodal Interaction (ICMI)*. 2016.

[3] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[4] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256. URL: https://proceedings.mlr.press/v9/glorot10a.html.

[5] Andrew G. Howard et al. *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. 2017. arXiv: 1704.04861 [cs.CV]. URL: https://arxiv.org/abs/1704.04861.

[6] M Iqtait, F S Mohamad, and M Mamat. "Feature extraction for face recognition via Active Shape Model (ASM) and Active Appearance Model (AAM)". In: *IOP Conference Series: Materials Science and Engineering* 332.1 (Mar. 2018), p. 012032. DOI: 10.1088/1757-899X/332/1/012032. URL: https://dx.doi.org/10.1088/1757-899X/332/1/012032.

[7] Shan Li, Weihong Deng, and JunPing Du. "Reliable Crowdsourcing and Deep Locality-Preserving Learning for Expression Recognition in the Wild". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2584–2593. DOI: 10.1109/CVPR.2017.277.

[8] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2019. arXiv: 1711.05101 [cs.LG]. URL: https://arxiv.org/abs/1711.05101.

[9] Patrick Lucey et al. "The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*. 2010, pp. 94–101. DOI: 10.1109/CVPRW.2010.5543262.

[10] Jia Le Ngwe et al. "PAtt-Lite: Lightweight Patch and Attention MobileNet for Challenging Facial Expression Recognition". In: *IEEE Access* 12 (2024), pp. 79327–79341. DOI: 10.1109/ACCESS.2024.3407108.

[11] Zeinab Sedaghatjoo, Hossein Hosseinzadeh, and Bahram Sadeghi Bigham. *Local Binary Pattern(LBP) Optimization for Feature Extraction*. 2024. arXiv: 2407.18665 [cs.CV]. URL: https://arxiv.org/abs/2407.18665.

[12] Ning Zhang, Junmin Luo, and Wuqi Gao. "Research on Face Detection Technology Based on MTCNN". In: *2020 International Conference on Computer Network, Electronic and Automation (ICCNEA)*. 2020, pp. 154–158. DOI: 10.1109/ICCNEA50255.2020.00040.

[13] Yuxiao Zhou and Kecheng Yang. "Exploring TensorRT to Improve Real-Time Inference for Deep Learning". In: *2022 IEEE 24th Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*. 2022, pp. 2011–2018. DOI: 10.1109/HPCC-DSS-SmartCity-DependSys57074.2022.00299.