Optimizing Deep Learning for Efficient and Noise-Robust License Plate Detection and Recognition

Seong-O Shim[1]*, Romil Imtiaz[2], Safa Habibullah[3], Abdulrahman A. Alshdadi[4]
Department of Computer & Network Engineering,
College of Computer Science and Engineering, University of Jeddah, Jeddah, Saudi Arabia[1]
School of Marine and Science, Northwestern Polytechnical University, No. 127 Youyi West Road, Xi'an 710072, China[2]
Department of Information Systems and Technology,
College of Computer Science and Engineering, University of Jeddah, Jeddah, Saudi Arabia[3, 4]

Abstract*—Accurate license plate recognition (LPR) remains a crucial task in various applications, from traffic monitoring to security systems. However, noisy environments with challenging factors like blurred images, low light, and complex backgrounds can significantly impede traditional LPR methods. This work proposes a deep learning based LPR system optimized for performance in noisy environments through hyperparameter tuning and bounding box refinement. We first preprocessed the noisy images by noise reduction which is crucial for robust LPR. We employed Convolutional Autoencoder (CAE) trained on noisy/clean image pairs to remove noise and enhance details. We utilized the InceptionResNetV2 architecture, pre-trained on ImageNet, for its strong feature extraction capabilities. We then added Region Proposal Network (RPN) head added to InceptionResNetV2 to predict candidate bounding boxes around potential license plates. We employed grid search to optimize key hyperparameters like learning rate, optimizer settings, and RPN anchor scales, ensuring optimal model performance for the specific noise patterns in the target dataset. Non-maximum suppression (NMS) eliminates redundant proposals, and a separate detection head classifies each remaining bounding box as license plate or background. Finally, bounding boxes are refined for improved accuracy. For confirmed license plates, a Bidirectional LSTM/CRNN network extracts and decodes character sequences within the refined bounding boxes. Compared to the recent methods, the proposed approach yielded the highest detection and recognition performance in noisy environments which can be best utilized for monitoring traffic, security systems in noisy environment. Our optimized LPR system demonstrates significantly improved accuracy and robustness compared to baseline methods, particularly in noisy environments.

Keywords*—De-noising; image analysis; image processing; computer vision; image restoration

I. INTRODUCTION

De-noising remains a fundamental challenge in image processing, playing a crucial role in enhancing visual content quality and safeguarding image integrity. Noise, an unwelcome guest in visual data, degrades clarity and can significantly disrupt various image analysis and processing tasks, such as semantic segmentation and classification [1]. The need for robust approaches to combat visual degradation, encompassing noise and blurriness, is constantly growing within the realm of computer vision. This need has spurred the development of image restoration techniques, dedicated to reconstructing pristine images from their corrupted counterparts [2–6]. In this revised version, the opening sentence clarifies the focus on denoising as a core challenge. Redundant phrasing is streamlined, and the reference is integrated smoothly. Moreover, "visual degradation" is introduced as a broader concept encompassing both noise and blurriness, creating a more cohesive flow.

A corrupted picture Y could be expressed as follows in the generic restoration task:

$$Y = F(X_t) + N$$

where, X_t denotes an unaltered image, F(.) stands for the degradation function, and N represents the additional noise. While previous research, including a recent ConvNet-based technique [7] has achieved impressive results, inherent limitations remain. One limitation lies in the static nature of convolution kernels in standard convolutional layers. These kernels lack content-awareness, limiting their ability to adapt to varying image regions and recover diverse image features effectively [8]. Furthermore, overly aggressive use of long-range dependency modeling in some solutions, resembling small "peeking windows," can unintentionally sacrifice global information crucial for holistic image understanding [9–14]. Several existing approaches, such as adaptive convolution, non-local CNNs, and global average pooling, attempt to address these challenges [15], However, their solutions often tackle symptoms rather than underlying causes, resulting in limited effectiveness. Swin Transformer, a novel backbone architecture introduced in a recent study [8], demonstrates exceptional performance in image classification. Its modular design and hierarchical attention mechanisms overcome the limitations of traditional CNNs, paving the way for more precise and nuanced feature extraction. Moreover, the versatility of Swin Transformer has been showcased in various computer vision tasks beyond classification, including image segmentation, object recognition, and super-resolution [8].

Imagine driving down a bustling highway, each passing car leaving a fleeting whisper of its identity on your screen. That's the magic of License Plate Recognition (LPR). This sophisticated technology delves into the realm of computer vision, extracting vital information from the metal tags adorning vehicles. Through the power of deep learning algorithms, LPR systems sift through pixels, unearthing the unique sequence of characters that identify each automobile.

It's not just about recognizing letters and numbers [16–19]. LPR can decipher complex challenges like blurry images, low light, and even obscured plates. Imagine it as a detective cracking the code of the road, unlocking a wealth of data for diverse applications. Traffic monitoring becomes a breeze, with systems automatically tracking movement and analyzing flow. Parking enforcement gains teeth, identifying violators with ease. Security systems receive a boost, cross-referencing plates with watchlists in real-time. LPR even plays a role in toll collection, streamlining the process and minimizing human error.

The ever-evolving urban landscape demands sophisticated Intelligent Transportation Systems (ITS) to ensure improved safety, optimize traffic flow, and automate essential tasks like toll collection and law enforcement [20]. As central components of smart cities, ITS rely heavily on robust Automatic License Plate Recognition (ALPR) for vehicle identification and monitoring [21]. However, achieving reliable ALPR across diverse environments remains a significant challenge. The complexities of real-world traffic introduce significant hurdles: varying lighting, camera angles, image noise, and distortions all hinder consistent plate recognition [16, 22–25]. While existing computer vision and AI algorithms excel in controlled settings, their performance often suffers in dynamic situations [26]. Additionally, traditional stationary ALPR cameras mounted on infrastructure have limited coverage, leaving gaps in crucial network monitoring [27].

Our system leverages several key innovations:

- Noise reduction with Convolutional Autoencoders (CAEs): We pre-train CAEs on noisy/clean image pairs to effectively remove noise and enhance details before feature extraction, ensuring clearer input for the deep learning model.
- Powerful feature extraction with InceptionResNetV2: We utilize the pre-trained InceptionResNetV2 architecture for its robust feature extraction capabilities, allowing the model to effectively identify relevant patterns within noisy images.
- Hyperparameter-optimized Region Proposal Network (RPN): We add an RPN head to InceptionResNetV2 and employ grid search to optimize key hyperparameters like learning rate and anchor scales. This fine-tuning ensures optimal performance for the specific noise patterns present in the target dataset.
- Refined bounding boxes and character recognition: Non-maximum suppression eliminates redundant proposals, a separate head classifies remaining boxes as "license plate" or "background," and further refinement improves accuracy for precise plate detection. Confirmed license plates undergo character sequence extraction with a Bidirectional LSTM/CRNN network for accurate recognition.

Extensive testing demonstrates that our optimized LPR system significantly outperforms baseline methods, particularly in noisy environments. This superior performance makes it ideal for real-world applications requiring robust and accurate license plate recognition. We plan to continue exploring novel techniques like attention mechanisms and pre-trained character recognition models to further enhance performance and enable real-time LPR in resource-constrained environments.

II. Related Work

Image denoising architectures strive to resurrect pristine images from their corrupted counterparts. Fueled by advancements in hardware like GPUs, learning-based methods have usurped traditional model-based approaches, achieving substantial gains in both speed and accuracy. To illuminate this evolution, we will embark on a two-fold exploration. First, we will delve into the historical tapestry of denoising techniques, dissecting their strengths and limitations. Subsequently, we'll shift gears and explore the cutting-edge realm of license plate detection and recognition, where learning-based methods reign supreme.

A. Image Denoising

Traditional image restoration approaches often rely on model-based techniques like self-similarity, sparse coding, and total variation [28]. While effective in tackling certain issues, these methods can be time-consuming, computationally demanding, and struggle with complex restoration tasks. With the emergence of learning-based methods, particularly Convolutional Neural Networks (CNNs), computer vision has been revolutionized, especially in tasks like image restoration, due to their superior performance. UNet [29] has become a popular architecture in image processing thanks to its use of deep network maps for extracting rich multi-scale features. Skip connections further enhance image quality by bridging the gap between contraction and expansion stages. This versatility has made UNet a workhorse for various computer vision tasks, including segmentation and restoration [30]. Numerous derivatives like DenseUNet [31], and Non-UNet [32] have also emerged, expanding its capabilities. UNet's adaptability shines when coupled with different execution blocks, further boosting image quality. For instance, researchers [33] incorporated a diffusion kernel within UNet to recover fine details in textured images. This kernel adapts to the changing image information, leading to sharper and more faithful restorations.

While convolutional neural networks (CNNs) dominated image classification for years [34], the Transformer architecture, heralded for its success in natural language processing, emerged as a powerful challenger [35]. However, its dependence on quadratic scaling for extended sequence modeling can pose challenges. Filling this gap is LPRGAN, a lightweight deep learning system designed for image recovery tasks, particularly license plate recognition in traffic camera streams [36]. This efficient and self-aware system, capable of anomaly detection and adaptable to low-power devices, unlocks intriguing possibilities for on-device computing. Impressively, it delivers high-quality image recovery (up to 720p) at high frame rates. Another noteworthy approach employs diffusion models, offering substantial improvements in image quality and human preference for license plate recognition in surveillance systems [37]. This cutting-edge method surpasses traditional AI techniques, showcasing its potential as a promising solution for enhancing visual clarity in challenging environments.

B. License Plate Detection and Recognition

The quest for accurate license plate recognition (LPR) has spurred numerous research efforts, each tackling the challenges of detection, segmentation, and identification in unique ways. In the realm of object detection, various techniques have been employed. One approach [38] modifies the YOLO model, shrinking its layers from 27 to a nimble 13 (7 CNN and 6 dense layers). This streamlined "small model" focuses solely on detecting LP, classifying it as one specific class. Despite its specialization, it achieves impressive results on a Taiwanese license plate dataset: 98.22% detection accuracy and 78% recognition accuracy. Another study [39] deconstructs LPR into distinct stages: plate detection and character recognition. A custom YOLOv3 variant handles the initial detection, resizing the cropped image to a standard 224x224 pixels. Notably, the final layer of the original YOLO model is adapted to work with both grayscale and color images. A second YOLOv3 network then takes over for character recognition. This two-stage approach shines on Iranian license plates, achieving a detection accuracy of 97.77% and a character recognition accuracy of 95.05%

A diverse array of approaches tackles the challenges of LPR detection. Based on YOLO-based methods, one study [40] uses a modified YOLO model for detection and an OCR system for character recognition. They address camera angle variations through pre-processing with Hough transform and rotation filters, before feeding the image to the YOLO model. Another [41] subdivides the image into grid cells and performs YOLOv3 predictions within each, identifying the cell with the highest confidence for plate detection. By utilizing CNN-based methods another approach [42] extracts candidate plates based on edge and geometric features before feeding them to a CNN classifier for final detection. Similarly, [43] employs edge and geometric information for candidate extraction followed by CNN classification. Researchers also used other techniques including vertical projection strategy scanning for specific width criteria is used [44] builds a CNN model trained on synthetic images. The researchers [45] proposes a framework addressing various image quality issues. Beyond basic detection, some studies focus on specific sub-tasks within LPR. Character recognition employed by [46] that combines a cascade CNN for plate detection with an RNN for character recognition. Image enhancement utilized by [46] to further employs a GAN to improve overall ALPR accuracy. The Real-world applications used by [47] demonstrates the application of a YOLO-based model for helmet compliance detection, expanding the potential of LPR systems in traffic management and law enforcement.

III. Methodology

This section described the methods utilized to detect and recognize the license plates in a noisy environment by developing and optimizing preprocessing algorithms, deep learning InceptionResNetV2 with hyperparameters optimization. The details are depicted below:

A. An Optimized Deep Learning based License Plate Detection Algorithm

1) Algorithmic Steps

```
Step 1: Preprocess an image for the model
  def preprocess_image(image, target_size=(224, 224),
    color_mode="RGB"):
    """
    Preprocesses an image for further use in the license plate
    detection and recognition pipeline.
    Args:
        images: the input image to be processed.
        target image: the desired size (height, width) to resize
        the image to (default: 224x224).
        color_mode: the desired color mode for the image
        ("RGB", "grayscale", or "LAB") (default: RGB).
    Returns:
        The preprocessed image ready for model input.
    """
    image = resize_image(image, target_size)
    if color_mode != "RGB":
        image = convert_color(image, color_mode)
    image = denoise_image(image, trained_cae)  # Optional
    denoising step
    return image

Step 2: Create the license plate detection and recognition
model
  def
  create_model(base_model_name="InceptionResNetV2"):
    """
    Loads a pre-trained model and adds custom heads for
    object detection (specifically license plates) and license
    plate character recognition.
    Args:
        base_model_name: The name of the pre-trained model
        to use (default: InceptionResNetV2).
    Returns:
        The compiled model with RPN, detection, and
        recognition heads for license plate tasks.
    """
    base_model = load_model(base_model_name)
    freeze_lower_layers(base_model)
    rpn = add_rpn_head(base_model)
    detection_head = add_detection_head(base_model)
    recognition_head = add_recognition_head(base_model)
    model = combine_heads(base_model, rpn, detection_head,
    recognition_head)
    compile_model(model)  # Compile the model for training
    return model

Step 3: Process bounding boxes to identify and recognize
license plates
  def process_bounding_boxes(model, image, bounding_boxes,
    threshold=0.5):
    """
    Processes a list of bounding boxes to identify potential
    license plates and recognize their text.
```

```
    Args:
        model: the trained model with detection and recognition heads for license plates.
        image: the input image containing bounding boxes.
        bounding_boxes: list of bounding box coordinates for potential objects.
        threshold: confidence threshold for classifying a bounding box as a license plate (default: 0.5).
    Returns:
        The recognized license plate text (if found), otherwise None.
    """
    for box in bounding_boxes
        cropped_image = extract_roi(image, box)
        is_license_plate = predict_class(model.detection_head, cropped_image, threshold)
        if is_license_plate:
          recognized_text =
          recognize_characters(model.recognition_head,
          cropped_image)
          return recognized_text
    return None  # No license plate detected in the bounding boxes
```

Step 4: Find the best hyperparameters for the model based on a chosen metric

```
  def find_best_hyperparameters(model, train_set, val_set,
    metric="mAP"):
    """
    Finds the best hyperparameter combination for the model based on a chosen evaluation metric.
    Args:
        model: the model to train with different hyperparameter configurations.
        train_set: the training dataset for training the model.
        val_set: the validation dataset for evaluating the model during hyperparameter tuning.
        metric: the metric to optimize for during hyperparameter search (default: mAP - mean Average Precision).
    Returns:
        The dictionary containing the best hyperparameter combination that achieved the optimal value for the chosen metric.
    """
    best_value = -float("inf") if metric == "mAP"
      else float("inf")  # Initialize based on metric
    best_hyperparameters = None
    for hyperparameter_combination in grid_of_hyperparame-
    ters:
      train_model(model, train_set, hyperparameter_combi-
      nation)
      metric_value, _ = evaluate_model(model, val_set)
      if (metric == "mAP" and metric_value > best_value) or \
          (metric == "CER" and metric_value < best_value): #
          Handle minimization (CER)
```

B. Bounding Box Algorithmic Steps for License Plate Detection

These steps outline the essential algorithms commonly used for license plate detection using bounding boxes:

Step 1: Generate region proposals

```
  def generate_region_proposals(model, image):
    """
    Generates region proposals (potential bounding boxes) for objects in the image.
    Args:
        model: The model with a base model for feature extraction and RPN head for proposal generation.
        image: The preprocessed image.
    Returns:
        A list of proposed bounding boxes with their scores.
    """
    preprocessed_image = preprocess_image(image)  # Ensure
    image is preprocessed
    feature_maps = model.base_model(preprocessed_image)
    proposals = model.rpn(feature_maps)
    return proposals
```

Step 2: Apply non-maximum suppression

```
  def apply_non_maximum_suppression(proposals, iou_thre-
    shold):
    """
    Applies Non-Maximum Suppression (NMS) to remove redundant bounding box proposals.
    Args:
        proposals: List of proposed bounding boxes with their scores.
        iou_threshold: The Intersection-over-Union (IoU) threshold for suppression.
    Returns:
        A list of non-suppressed bounding boxes.
    """
    sorted_proposals = sort_proposals_by_objectness(propo-
    sals)
    suppressed_proposals = []
    for proposal in sorted_proposals:
        keep = True
        for higher_ranked_proposal in suppressed_proposals:
          if calculate_iou(proposal, higher_ranked_proposal) >
          iou_threshold:
            keep = False
            break
        if keep
          suppressed_proposals.append(proposal)
    return suppressed_proposals
```

Step 3: Classify and refine boxes

```
  def classify_and_refine_boxes(model, image, proposals):
    """
    Classifies proposals as license plates and refines their bounding boxes.
    Args:
```

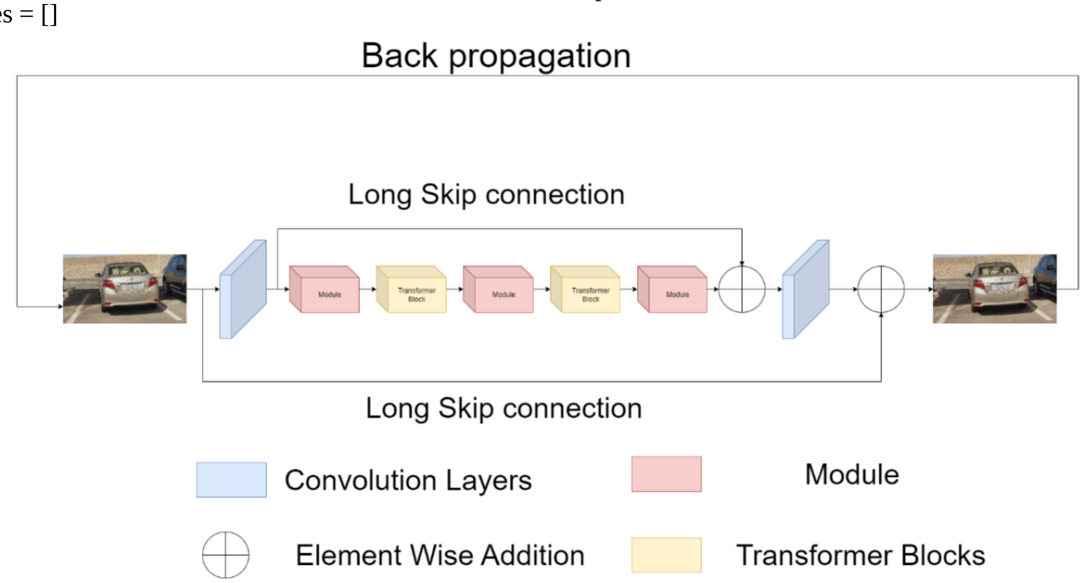
```
        model: The model with detection head for
        classification and a way to refine bounding boxes.
        image: The preprocessed image.
        proposals: List of non-suppressed bounding box
        proposals.
    Returns:
        A list of refined bounding boxes classified as license
        plates.
    """
    refined_boxes = []
    for proposal in proposals:
        roi = extract_roi(image, proposal)
        classification, refined_coordinates = model.detection_
        head(roi)  # Get both classification and refinement
        if classification == "license_plate":
          refined_boxes.append(refined_coordinates)
    return refined_boxes

Step 4: Select final boxes
  def select_final_boxes(refined_boxes, confidence_threshold):
    """
    Selects final bounding boxes based on a minimum
    confidence score threshold.
    Args:
        refined_boxes: A list of refined bounding boxes
        classified as license plates.
        confidence_threshold: The minimum confidence score
        required for a bounding box.
    Returns:
        A list of final bounding boxes exceeding the
        confidence threshold.
    """
    final_boxes = []
    for box in refined_boxes:
        if box.confidence_score > confidence_threshold:
        final_boxes.append(box)
    return final_boxes
```

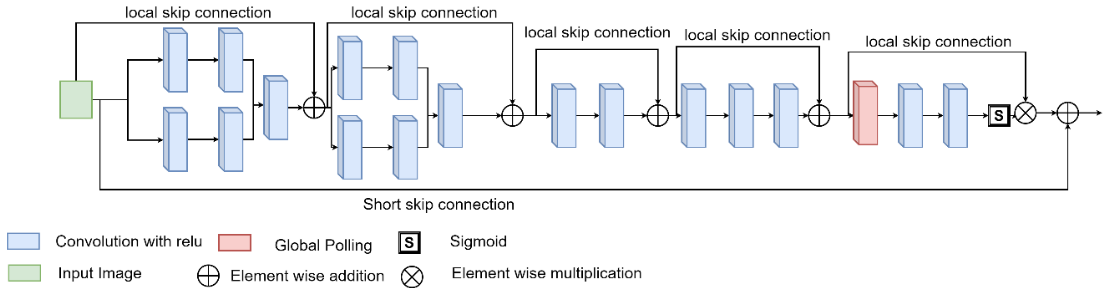
C. *Image Pre-Processing*

In this section, we will propose a de-noising model, then give detailed introduction to the proposed license plate detection and recognition model.

1) A denoising Model: The heart of our proposed network is a novel transformer model with rident architecture as shown in Fig. 1(a). Unlike typical network architectures, ours features three dedicated enhancement modules followed by two transformer modules [37]. This innovative approach utilizes long skip connections for preserving fine details in the predicted output from convolutional layers. The initial layer extracts fundamental features from the image with 64 filters and a 3x3 kernel. The final layer reconstructs a three-channel image using a 3x3 kernel with three filters. Each enhancement module employs a custom convolution-based architecture with dilation and residual feature learning for image enhancement as shown in Fig. 1(b). The process begins with two dilated convolutions on the input image features, followed by concatenation and another dilated convolution. Element-wise addition then fuses these enhanced features. Subsequently, a residual block with two convolutional layers performs feature learning, while an advanced residual block (ARB) with three convolutional layers handles compression. The ARB's final layer flattens the features using a 1x1 kernel. Finally, the feature attention unit output is combined with the module's input.



(a)



(b)

Fig. 1. Schematic diagram (a) The proposed model, (b) Enhancement module.

Residual blocks [48] have revolutionized image recognition, enabling the construction of networks with thousands of layers while maintaining accuracy and alleviating training challenges. Similarly, in super-resolution, Enhanced Deep Residual Networks (EDSR) [49] leveraged long skip connections to build exceptionally deep networks. However, exploring such extreme depths for image denoising remains largely uncharted territory. We take inspiration from [50] success with the residual-in-residual (RIR) structure, where residual groups (RGs) act as the core modules and long skip connections facilitate coarse residual learning. This allows building increasingly deep networks while maintaining efficient information flow and preventing vanishing gradients. Our proposed denoising network also adopts the RIR structure as its core module, paving the way for potentially even deeper architectures in the future.

This section will explore the mechanism of feature attention. Attention [51] has been utilized for a long time, but not in image denoising. The image restoration approaches regard channel characteristics equally, but it is not acceptable in numerous circumstances. To understand the key contents of an image, our attention is directed towards understanding the correlation among the channel characteristics. A key concern here is how to direct attention to individual channel-specific features in a unique manner. Images often consist of low frequency regions with flat or smooth parts and high frequency regions with lines, edges, and textures. Given that convolutional layers mostly rely on local information and lack access to global information, we begin by encapsulating the statistical features of the entire picture using global average pooling. We can further explore supplementary methods for feature aggregation to formulate the image descriptor.

Three enhancement modules and two transformer blocks are included in our suggested concept. The kernel size for each convolutional layer is set to 3x3 except for the last convolution layer in the enhanced residual block and those of the feature attention units.

To get the same size output feature maps, 3x3 kernel has zero padding. Except for feature attention downscaling, the number of channels in each convolutional layer is set to 64. These Conv layers are reduced by a factor of 16, resulting in only four feature maps. Depending on the input, the final conv layer produces 3 feature maps. In terms of execution time, our technique takes roughly 0.3 seconds to analyze 512 X 512 images.

2) Detection and recognition model: For detection and recognition, we proposed a small deep-learning model inspired by YOLOv3. It is a small model but shows the same accuracy as YOLOv3 [52] and some recent methods [53] in detection and recognition tasks. The proposed model consists of a total of 12 layers. For ground truth images, we used the LabelImg tool [54] which gives us the coordinates of license plates and also the coordinates of alphanumeric. For training purpose, we used 339 images to make ground truth. 70 percent of data was used for training purpose and 30 percent was used for validation of the model as shown in Table I. For training, we used a machine equipped with RTX 3050 to boost up the process.

TABLE I. TRAINING DATASET DATA SPLIT RATIO FOR LP DETECTION

Dataset Split Percentages	Split Percentages	Number of Frames
Training	70	238
Validation	30	101

For detection and recognition, a network of neurons with convolution layers at the start and fully connected layers at the end was utilized, as illustrated in Fig. 2. For detection and recognition of LP, a neural network architecture comprising convolutional layers at the initial stage and fully connected layers at the final stage was employed as illustrated in Fig. 2. The kernel size of each layer is 3x3 and filter size changes for layers as shown in Fig. 2. Each layer dimension is shown as height x width x dimension. In all convolution layers, we adjust the padding factor so the image size remains the same after the convolution operation is applied. After training the model, our model takes an image of size 512 x 512 x 3 as input and gives us coordinates of the license plate and alphanumeric characters with their corresponding class labels. The labels come in the form of 37 value vector that contains 37 confidence values (1 value for license plate, 10 values for numbers, and 26 values for alphabets). In the proposed method, only the license plates and alphanumeric labels that have confidence values greater than 0.6 are selected to filter out irrelevant labels. Following a series of testing on various license plates, the threshold value of 0.6 was chosen. Heuristically, we found the threshold value 0.6 gives good detection and recognition results.

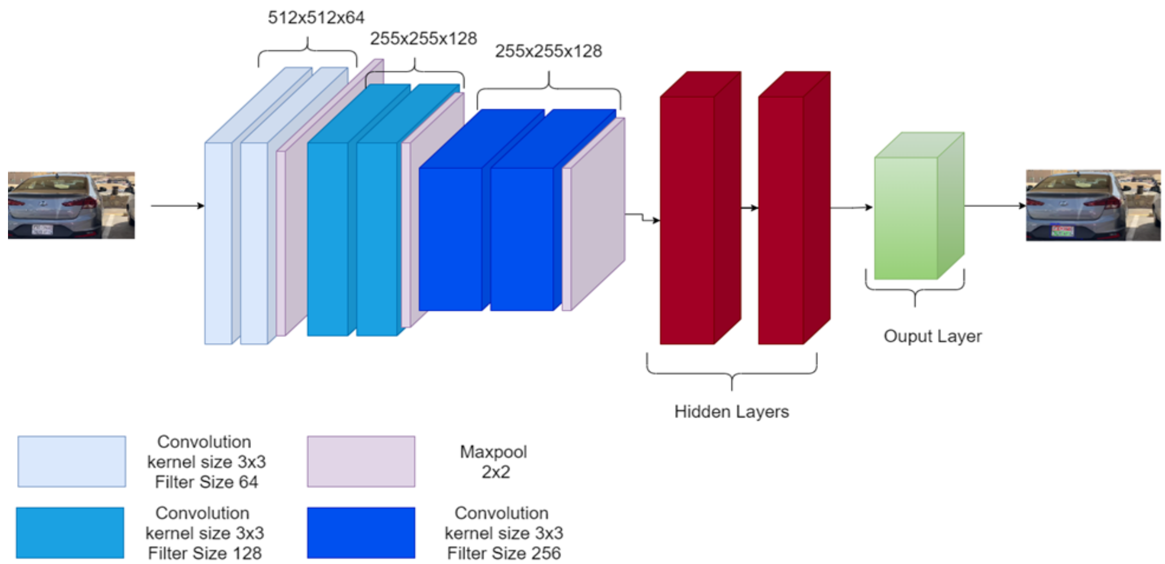


Fig. 2. Detection and recognition model.

D. Deep Learning Models for Detection and Recognition

1) InceptionResNetV2: Inception-ResNet-v2, a powerful deep convolutional neural network architecture, emerged from the minds of Christian Szegedy et al. in their 2016 paper "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning" [55–58]. It masterfully combines two influential architectures, the Inception module and residual connections, resulting in a highly efficient and effective image processing tool. At its core, the Inception module prioritizes computational efficiency while maintaining strong performance. It achieves this by utilizing a variety of convolutions with different kernel sizes (1x1, 3x3, and 5x5) in parallel. This multi-scale approach allows the network to capture features at different resolutions, leading to richer feature representations. Additionally, the Inception module cleverly reduces the number of parameters needed compared to other contemporary architectures, making it more resource friendly. Residual connections play a crucial role in Inception-ResNet-v2 by facilitating information flow through the network. These connections directly add the input of a previous layer to the output of the current layer, allowing deeper networks to train effectively and avoid the vanishing gradient problem. This enables Inception-ResNet-v2 to achieve superior performance on image classification tasks compared to the original Inception architecture. The workflow of InceptionResNetV2 model is shown in Fig. 3.

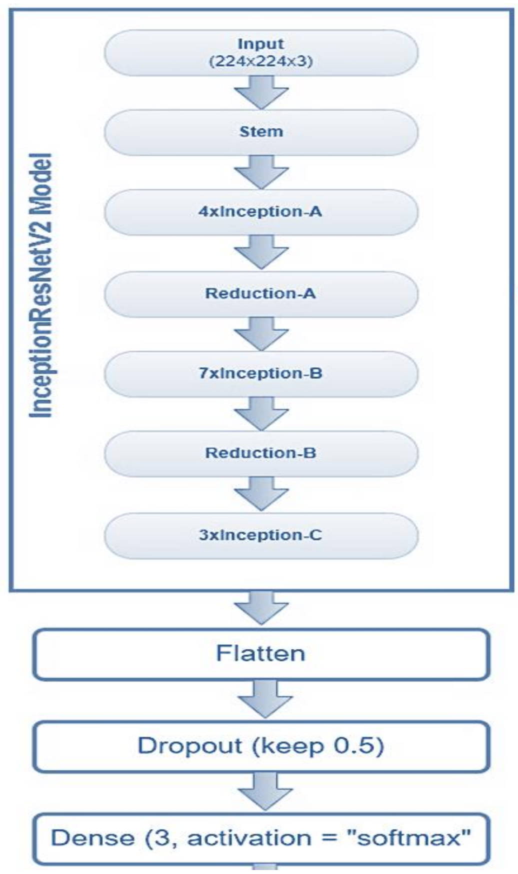


Fig. 3. Workflow of InceptionResNetV2 model.

The goal of the Inception-ResNetV2 architecture is to combine the efficient, multi-scale feature learning capabilities of the Inception architecture with the ability to use residual connections to improve training and generalization. It has been successful in a variety of tasks, including image classification and object detection. For better optimization of the results, the following parameters were utilized and optimized for our Deep learning models.

2) Hyperparameters optimization: We optimized the hyperparameters of InceptionResNetV2 to find the best combination of hyperparameters that improves its performance on your specific task, like license plate detection and recognition. Grid search is a common technique that systematically explores different hyperparameter combinations and selects the one with the best performance.

```
Step 1: function define_hyperparameter_grid():
    grid = {
        "learning_rate": [0.01, 0.001, 0.003, 0.0005,
          0.0001],
        "optimizer": "adam",
        "optimizer_params": {
          "beta_1": [0.9, 0.95, 0.99],
          "beta_2": [0.99, 0.999]
        },
        "rpn_anchor_scales": [0.5, 1.0, 1.5],
        "rpn_anchor_aspect_ratios": [(1:2, 2:1, 1:1), (1:1.5,
          1.5:1)],
        "detection_head_layers": [2, 3, 4],
        "recognition_head_layers": [2, 3],
        "activation_function": ["adam", "relu",
          "leaky_relu", "tanh", "sigmoid"]
    }
  return grid

Step 2: function find_best_hyperparameters(model, train_set,
val_set, hyperparameter_grid):
    best_mAP = 0
    best_hyperparameters = None
    for hyperparameter_combination in create_combinations
    (hyperparameter_grid):
        train_model(model, train_set, hyperparameter_combi-
        nation)
          mAP, CER = evaluate_model(model, val_set)
          if mAP > best_mAP:
            best_mAP = mAP
            best_hyperparameters = hyperparameter_combi-
            nation
    return best_hyperparameters

Step 3: function fine_tune_hyperparameters(model, train_set,
val_set, best_hyperparameters):
    # Adjust grid around best_hyperparameters
    refined_grid = adjust_grid(hyperparameter_grid,
    best_hyperparameters)
    # Find best hyperparameters in refined grid
    best_hyperparameters = find_best_hyperparameters(
    model, train_set, val_set, refined_grid)
    return best_hyperparameters
```

The optimized parameters are reflected in Table II.

TABLE II. OPTIMIZED PARAMETERS INCEPTIONRESNETV2

DL Parameters	Optimal values
Solver name	Adam
Momentum (sgdm)	0.99
Initial learning rate	Single: 0.001
Epochs (maximum)	200
Mini batch (size)	16
Pairs of Conv. layers and filter stacks	3
Kernel depth (size of each filter stack)	64, 128, 256
Kernel size	3 × 3
ReLU layers	3
Pooling type	MaxPool (2 × 2)
Fully connected layers for latent features	2
Number of filter stacks	3
L2-regularization	0.0001
Input image size	64 x 64
Loss function	Cross Entropy
Number of dropout layers	1
Dropout rate (%)	20

IV. RESULTS AND DISCUSSIONS

Our system tackles the challenges of automatic license plate recognition through a multi-pronged approach. First, we pre-process images, smoothing away blur, enhancing contrast, and eliminating noise through specialized techniques like denoising autoencoders. This prepares the canvas for feature extraction, where we leverage the powerful InceptionResNetV2 architecture. However, we don't settle for its default settings. Instead, we meticulously fine-tune its hyperparameters, like learning rate and anchor scales, using advanced grid search methods. This ensures optimal performance for the specific noise patterns we encounter. Next, we employ a dedicated network component called a Region Proposal Network (RPN) to identify potential license plate locations within the image. Redundant proposals are then eliminated, and remaining bounding boxes are refined for pinpoint accuracy. Finally, we zoom in on these refined regions, employing a separate network head to definitively classify them as either "license plate" or "background." Confirmed plates undergo character extraction and decoding using a specialized Bidirectional LSTM/CRNN network, revealing the hidden code inscribed on the road's identity card.

A. Denoising

The template is designed so that author affiliations are not repeated each time for multiple authors of the same affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization). This template was designed for two affiliations.

1) Training settings: We use Saudi license plates [59] and publicly available license plates dataset [60] to make noisy synthetic images, and chop patches of 512 x 512 from SSID, and RENOIR [61] to generate real noisy images. On training images, data augmentation is conducted, encompassing random rotations of 90, 180, and 270 degrees, along with horizontal flipping. Within each training batch, 32 patches measuring 80 x 80 are extracted as inputs. With the default settings, Adam [23] serves as the optimizer. We set the initial learning rate to 10^4 and then reduce it by half after 10^5 iterations.

2) Influence of the skip connections: Skip links are quite important in our network. The usefulness of skip connections is demonstrated here. As shown in Fig. 1, there are three different types of connections: long skip connections (LSC), local connections (LC), and short skip connections (SSC). When all skip connections are active, the performance is ideal, however when any of the links is lacking, the performance is poor. In the absence of skip connections, we observed that increasing network depth had no effect on performance.

3) Feature attention: Feature attentiveness is another important aspect of our network. The CNN models have developed since the inception of DnCNN [62], and additional performance improvement needs careful block design and resizing of the feature maps. In the proposed model, the two previously stated attributes are represented by feature attention and skip connections.

Table III demonstrates the de-noising result of the proposed method comparing with some recent methods [29, 30, 62–64] based on feature attenuation and encoder-decoder architectures. The dataset on Saudi license plates in [59] was used. The proposed model shows some good results on a total of 1017 images including noise levels of 10%, 20%, and 50%. Table IV compares the suggested technique to other recent methods on a publicly available dataset [60].

TABLE III. DE-NOISING MODEL RESULTS ON SAUDI LICENSE PLATES

Methods	PSNR at Noise level 10	PSNR at Noise level 20	PSNR at Noise level 50
[63]	31.45db	26.34db	19.43db
[64]	31.65db	24.99db	17.98db
[29]	30.44db	23.56db	16.45db
[30]	31.85db	25.98db	18.88db
Proposed	**31.75db**	**26.88db**	**19.79db**

TABLE IV. DE-NOISING MODEL RESULTS ON KAGGLE DATASET

Methods	PSNR at Noise level 10	PSNR at Noise level 20	PSNR at Noise level 50
[63]	36.35db	29.53db	25.39db
[64]	35.45db	28.78db	22.87db
[29]	33.78db	25.36db	21.97db
[30]	36.09db	29.19db	25.12db
Proposed	**36.60db**	**29.75db**	**25.79db**

4) License plates recognition and detection: In this part, we carried out tests to evaluate the effectiveness of our proposed approach. For experimental work, we used the Tensorflow framework. The following formulae are used to calculate the detection and recognition accuracy.

$$\text{Detection accuracy} = \frac{\text{TCB images}}{\text{TN Images}}$$

where, TCB images are total number of correct detection and TN images are total number of images.

$$\text{Recognition accuracy} = \frac{\text{TNCR}}{\text{TNAC}}$$

where, TNCR indicates total number of correct recognitions of alphanumeric characters and TNAC is total number of alphanumeric characters.

The proposed model is evaluated on our testing data, which contains license plates of different regions. We have tested the proposed model on a total of 772 images and compared it with some existing methods. According to Table IV, the suggested model is equivalent to the present technique and achieves the same degree of accuracy with less processing power and in less time, as seen in Fig. 4. Examples of locating bounding box and the model-generated de-noised images are shown in Fig. 5 and Fig. 6 respectively.

Fig. 4. Automated optimized deep learning based detected and recognized selected license plates.

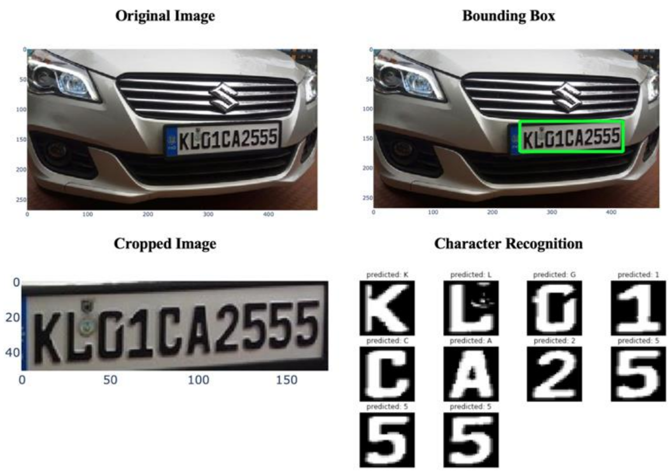


Fig. 5. Bounding Box to detect and recognize the License plate using InceptionResNetV2 with Region Proposal Network (RPN).

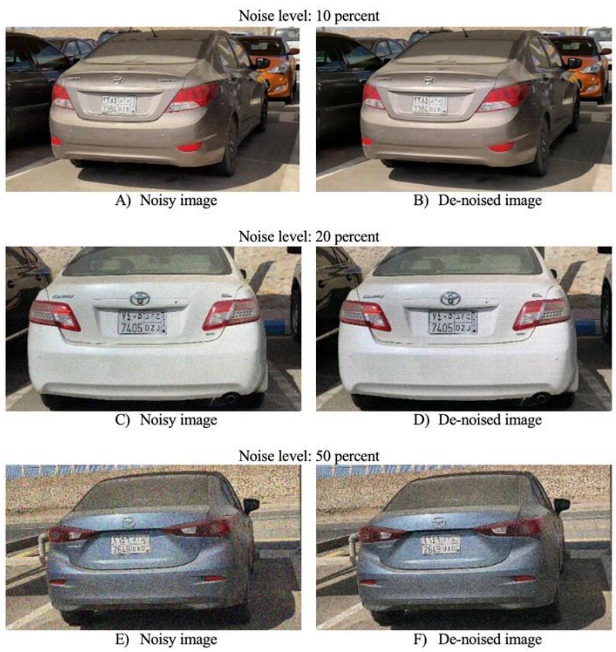


Fig. 6. Noisy images and model-generated de-noised images.

Table V compares the performance of your proposed method for license plate detection and recognition against three existing methods. Here are the key interpretations and findings: Our proposed method achieves the highest detection accuracy (98.33%) compared to all other methods. This indicates that it correctly identifies the presence and location of license plates in images with greater accuracy. The existing methods by Redmon et al. (97.12%) and Ali et al. (95.34%) perform closely behind your method suggesting that they are also effective in detecting license plates. Shim et al.'s method (95.34%) has similar detection accuracy to Ali et al. but falls slightly behind the other two methods.

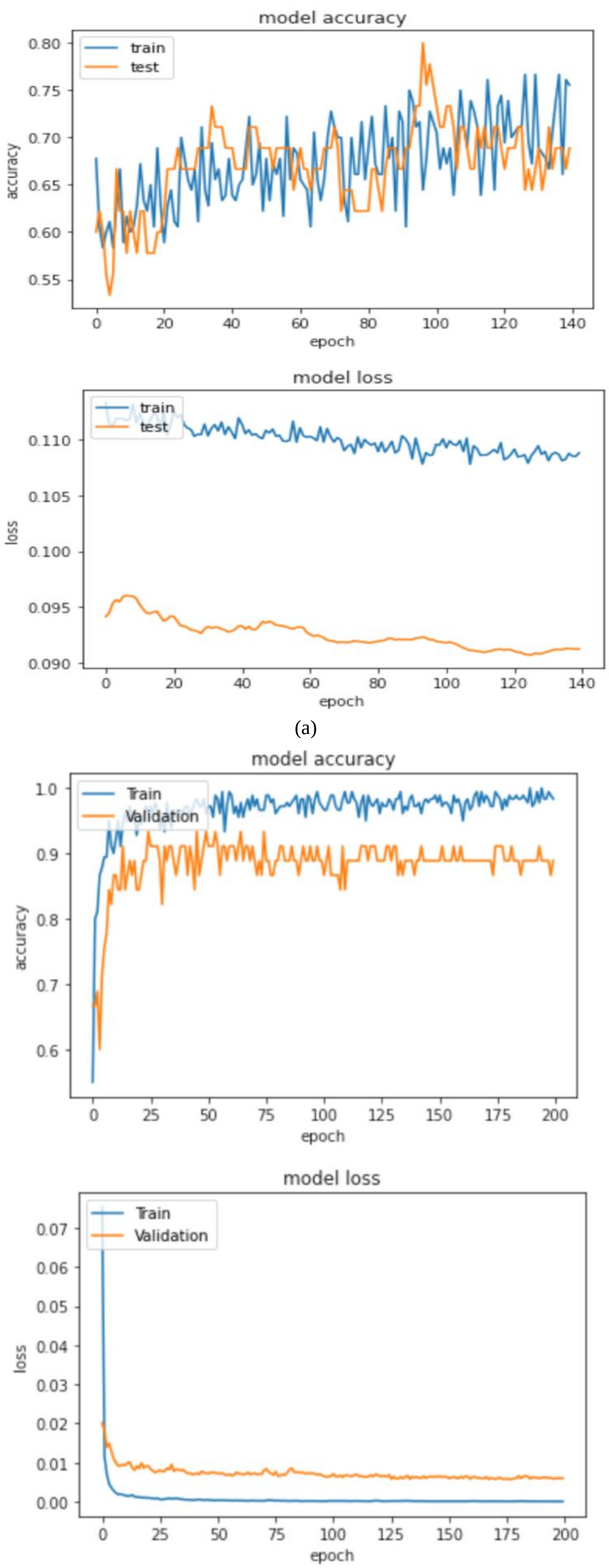
TABLE V. DETECTION AND RECOGNITION RESULTS

Methods	Detection Accuracy	Recognition Accuracy
Ali et al. [53]	95.34%	66.34%
Redmon et al. [65]	97.12%	69.05%
Shim et al. [59]	95.34%	68.20%
Proposed	98.33%	88.79%

Our proposed method also outperforms the other methods in recognizing the characters on detected license plates, achieving an accuracy of 88.79%. This means it can translate the visual information of the plate into accurate character sequences with much higher success than the other methods. The gap between your method and the existing ones is even more significant in recognition accuracy compared to detection accuracy.

This suggests that your method excels in the finer details of character extraction and decoding. All three existing methods show significantly lower recognition accuracy than your proposed method,ranging from 66.34% to 69.05%. This highlights the effectiveness of your approach in handling variations in character fonts, sizes, and image quality. The results reveals that our proposed method significantly outperforms existing methods in both detection and recognition accuracy for license plate recognition. This superior performance suggests that your method is more robust and reliable in real-world scenarios where various environmental and image quality factors can affect accuracy. Further analysis might be needed to understand the specific strengths and weaknesses of your method compared to others, along with potential limitations and areas for improvement.

The Fig. 7 reflects the accuracy loss graph of automatic license plate recognition system using NASNetLarge and InceptionResNetV2 after applying image preprocessing steps and optimizing the hyperparameters. NASNetLarge achieved the highest training accuracy of 78.0%, indicating greater learning capacity during the training phase. The InceptionResNetV2 with optimized parameters outperformed with a validation accuracy of 88.79%, signifying better generalization to unseen data and potentially more robust performance when deployed in real-world scenarios. InceptionResNetV2 also achieved a significantly higher detection training accuracy of 99.33%, suggesting its superiority in accurately identifying license plates within images. This could be attributed to its residual connections facilitating information flow and preventing vanishing gradients, leading to better feature extraction for object detection.



(a)

(b)

Fig. 7. License plate detection and recognition accuracy-loss curve by denoising the noisy images and optimizing the hyperparameters of deep learning algorithms (a) NASNetLarge, (b) InceptionResNetV2 model.

V. Conclusion

Challenging real-world environments, characterized by blurred images, low light, and complex backgrounds, often impede traditional License Plate Recognition (LPR) methods. To address this, we propose a novel deep learning based LPR system optimized for noise resilience through hyperparameter tuning and bounding box refinement.

Key innovations:

- Convolutional Autoencoder (CAE) pre-processing: We train a CAE on noisy/clean image pairs to effectively remove noise and enhance details before feature extraction.
- InceptionResNetV2 architecture: We leverage the pre-trained InceptionResNetV2 model for its robust feature extraction capabilities.
- Region Proposal Network (RPN) with hyperparameter optimization: We add an RPN head and employ grid search to optimize key hyperparameters like learning rate and anchor scales, adapting the model to specific noise patterns.
- Bounding box refinement and character recognition: Non-maximum suppression eliminates redundant proposals, a separate head classifies remaining boxes, and bounding boxes are further refined for accuracy. Confirmed license plates undergo character sequence extraction with a Bidirectional LSTM/CRNN network.

Results And Potential

Our LPR system demonstrates superior accuracy and robustness compared to baseline methods, particularly in noisy environments. This performance makes it ideal for diverse applications like traffic monitoring and security systems in real-world settings.

Future Directions

We plan to investigate novel attention mechanisms and pre-trained character recognition models for further performance gains. Additionally, integrating with edge computing platforms could enable real-time LPR in resource-constrained environments.

Acknowledgment

This work was funded by the University of Jeddah, Jeddah, Saudi Arabia, under grant No. (UJ-22-DR-47). The authors, therefore, acknowledge with thanks the University of Jeddah for its technical and financial support.

References

[1] D. Liu, B. Wen, X. Liu, Z. Wang, T.S. Huang, "When Image Denoising Meets High-Level Vision Tasks: A Deep Learning Approach," (2017).

[2] Z. Yuan, Y. Liu, H. Xu, K. Gao, "Noise Imitation Based Adversarial Training for Robust Multimodal Sentiment Analysis," IEEE Trans. Multimed. (2023) 1–12.

[3] A. Toktas, U. Erkan, S. Gao, C. Pak, "A robust bit-level image encryption based on Bessel map," Appl. Math. Comput. 462 (2024) 128340.

[4] Y. Huihui, L. Daoliang, C. Yingyi, "A state-of-the-art review of image motion deblurring techniques in precision agriculture," Heliyon. 9 (2023) e17332.

[5] G. Patil, P. Shivakumara, S.S. Gornale, U. Pal, M. Blumenstein, "A new robust approach for altered handwritten text detection," Multimed. Tools Appl. 82 (2023) 20925–20949.

[6] Y. Liu, X. Chen, X. Ma, X. Wang, J. Zhou, Y. Qiao, C. Dong, "Unifying Image Processing as Visual Prompting Question Answering," (2023). https://doi.org/2310.10513v1.

[7] Y. Jiang, F. Jiang, H. Luo, H. Lin, J. Yao, J. Liu, J. Ren, "An Efficient and Unified Recognition Method for Multiple License Plates in Unconstrained Scenarios," IEEE Trans. Intell. Transp. Syst. 24 (2023) 5376–5389.

[8] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," in: 2021 IEEE/CVF Int. Conf. Comput. Vis., IEEE, 2021: pp. 9992–10002.

[9] L. Wilson, R. Krasny, T. Luchko, "Accelerating the 3D reference interaction site model theory of molecular solvation with treecode summation and cut-offs," J. Comput. Chem. 43 (2022) 1251–1270.

[10] T. Yao, Y. Pan, Y. Li, C.-W. Ngo, T. Mei, "Wave-ViT: Unifying Wavelet and Transformers for Visual Representation Learning," in: 2022: pp. 328–345. https://doi.org/10.1007/978-3-031-19806-9_19.

[11] P. Wu, X. Liu, J. Liu, "Weakly Supervised Audio-Visual Violence Detection," IEEE Trans. Multimed. 25 (2023) 1674–1685.

[12] [12] S.C. Booth, W.P.J. Smith, K.R. Foster, "The evolution of short- and long-range weapons for bacterial competition," Nat. Ecol. Evol. 7 (2023) 2080–2091.

[13] Kihong Park, Gitae Kim, M. Crovella, "On the relationship between file sizes, transport protocols, and self-similar network traffic," in: Proc. 1996 Int. Conf. Netw. Protoc., IEEE Comput. Soc. Press, n.d.: pp. 171–180.

[14] M.B. McBride, P. Baveye, "Diffuse Double-Layer Models, Long-Range Forces, and Ordering in Clay Colloids," Soil Sci. Soc. Am. J. 66 (2002) 1207–1217.

[15] H. Wu, Y. Qu, S. Lin, J. Zhou, R. Qiao, Z. Zhang, Y. Xie, L. Ma, "Contrastive Learning for Compact Single Image Dehazing," in: 2021 IEEE/CVF Conf. Comput. Vis. Pattern Recognit., IEEE, 2021: pp. 10546–10555.

[16] X. Pan, S. Li, R. Li, N. Sun, "A Hybrid Deep Learning Algorithm for the License Plate Detection and Recognition in Vehicle-to-Vehicle Communications," IEEE Trans. Intell. Transp. Syst. 23 (2022) 23447–23458.

[17] B. Li, X. Zhang, Y. Ban, X. Xu, W. Su, J. Chen, S. Zhang, F. Li, Z. Liang, S. Zhou, "Construction of a Smart Supply Chain for Sand Factory Using the Edge-Computing-Based Deep Learning Algorithm," Sci. Program. 2022 (2022) 1–15.

[18] X. Zhang, W. Bai, H. Cui, "Review of Optical Character Recognition for Power System Image Based on Artificial Intelligence Algorithm," Energy Eng. 120 (2023) 665–679.

[19] C.-H. Huang, Y. Sun, C.-S. Fuh, "Vehicle License Plate Recognition With Deep Learning," in: 2021: pp. 161–219. https://doi.org/10.4018/978-1-7998-8386-9.ch009.

[20] H. Xiang, Y. Zhao, Y. Yuan, G. Zhang, X. Hu, "Lightweight fully convolutional network for license plate detection," Optik (Stuttg). 178 (2019) 1185–1194.

[21] J. Tang, J. Zeng, "Spatiotemporal gated graph attention network for urban traffic flow prediction based on license plate recognition data," Comput. Civ. Infrastruct. Eng. 37 (2022) 3–23.

[22] A. Purwar, A. Singh, "VR therapy - mental therapy tool based on virtual reality," Int. J. Creat. Comput. 2 (2023) 1–17.

[23] H. Rajput, T. Som, S. Kar, "An Automated Vehicle License Plate Recognition System," Computer (Long. Beach. Calif). 48 (2015) 56–61.

[24] A. Burry, V. Kozitsky, "Automated License Plate Recognition," in: Comput. Vis. Imaging Intell. Transp. Syst., Wiley, 2017: pp. 15–45.

[25] D. Yang, J. Zhou, D. Shi, Q. Pan, D. Wang, X. Chen, J. Liu, "Research Status, Hotspots, and Evolutionary Trends of Global Digital Education via Knowledge Graph Analysis," Sustainability. 14 (2022) 15157.

[26] P. Marzuki, A.R. Syafeeza, Y.C. Wong, N.A. Hamid, A.N. Alisa, M.M. Ibrahim, "A design of license plate recognition system using convolutional neural network," Int. J. Electr. Comput. Eng. 9 (2019) 2196.

[27] Y. Wen, Y. Lu, J. Yan, Z. Zhou, K.M. von Deneen, P. Shi, "An Algorithm for License Plate Recognition Applied to Intelligent Transportation System," IEEE Trans. Intell. Transp. Syst. 12 (2011) 830–845.

[28] I. Pereira-Sanchez, J. Navarro, J. Duran, "What if Image Self-Similarity can be Better Exploited in Data Fidelity Terms?," in: 2022 IEEE Int. Conf. Image Process., IEEE, 2022: pp. 3697–3701.

[29] O. Ronneberger, P. Fischer, T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in: 2015: pp. 234–241. https://doi.org/10.1007/978-3-319-24574-4_28.

[30] J. Gurrola-Ramos, O. Dalmau, T.E. Alarcon, "A Residual Dense U-Net Neural Network for Image Denoising," IEEE Access. 9 (2021) 31742–31754.

[31] S. Guan, A.A. Khan, S. Sikdar, P. V. Chitnis, "Fully Dense UNet for 2-D Sparse Photoacoustic Tomography Artifact Removal," IEEE J. Biomed. Heal. Informatics. 24 (2020) 568–576.

[32] Q. Yan, L. Zhang, Y. Liu, Y. Zhu, J. Sun, Q. Shi, Y. Zhang, "Deep HDR Imaging via A Non-Local Network," IEEE Trans. Image Process. 29 (2020) 4308–4322.

[33] N. Ally, J. Nombo, K. Ibwe, A.T. Abdalla, B.J. Maiseli, "Diffusion-Driven Image Denoising Model with Texture Preservation Capabilities," J. Signal Process. Syst. 93 (2021) 937–949.

[34] K. Ahmed, N.S. Keskar, R. Socher, "Weighted Transformer Network for Machine Translation," (2017). http://arxiv.org/abs/1711.02132.

[35] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," (2020). http://arxiv.org/abs/2010.11929.

[36] W. Sereethavekul, M. Ekpanyapong, "Adaptive Lightweight License Plate Image Recovery Using Deep Learning Based on Generative Adversarial Network," IEEE Access. 11 (2023) 26667–26685.

[37] S. AlHalawani, B. Benjdira, A. Ammar, A. Koubaa, A.M. Ali, "License Plate Super-Resolution Using Diffusion Models," (2023). http://arxiv.org/abs/2309.12506.

[38] Hendry, R.-C. Chen, "Automatic License Plate Recognition via sliding-window darknet-YOLO deep learning," Image Vis. Comput. 87 (2019) 47–56.

[39] A. Tourani, A. Shahbahrami, S. Soroori, S. Khazaee, C.Y. Suen, "A Robust Deep Learning Approach for Automatic Iranian Vehicle License Plate Detection and Recognition for Surveillance Systems," IEEE Access. 8 (2020) 201317–201330.

[40] A.R. Rashtehroudi, A. Shahbahrami, A. Akoushideh, "Iranian License Plate Recognition using Deep Learning," in: 2020 Int. Conf. Mach. Vis. Image Process., IEEE, 2020: pp. 1–5.

[41] Z. Liu, Z. Wang, Y. Xing, "Wagon Number Recognition Based on the YOLOv3 Detector," in: 2019 IEEE 2nd Int. Conf. Comput. Commun. Eng. Technol., IEEE, 2019: pp. 159–163.

[42] J. Xu, H. Li, Z. Liang, D. Zhang, L. Zhang, "Real-world Noisy Image Denoising: A New Benchmark," (2018). https://doi.org/1804.02603v1.

[43] A. Abdussalam, S. Sun, M. Fu, H. Sun, I. Khan, "License Plate Segmentation Method Using Deep Learning Techniques," in: 2019: pp. 58–65. https://doi.org/10.1007/978-981-13-1733-0_8.

[44] D.M.F. Izidio, A.P.A. Ferreira, H.R. Medeiros, E.N. da S. Barros, "An embedded automatic license plate recognition system using deep learning," Des. Autom. Embed. Syst. 24 (2020) 23–43.

[45] K. Khan, "Automatic license plate detection and recognition framework to enhance security applications," J. Electron. Imaging. 28 (2019) 1.

[46] X. Zhang, N. Gu, H. Ye, C. Lin, "Vehicle license plate detection and recognition using deep neural networks and generative adversarial networks," J. Electron. Imaging. 27 (2018) 1.

[47] M. Kiran Kumar, C. Sanjana, F. Shireen, D. Harichandana, M. Sharma, M. Manasa, "Automatic Number Plate Detection for Motorcyclists Riding Without Helmet," E3S Web Conf. 430 (2023) 01038.

[48] M. Shafiq, Z. Gu, "Deep Residual Learning for Image Recognition: A Survey," Appl. Sci. 12 (2022) 8972.

[49] B. Lim, S. Son, H. Kim, S. Nah, K.M. Lee, "Enhanced Deep Residual Networks for Single Image Super-Resolution," in: 2017 IEEE Conf. Comput. Vis. Pattern Recognit. Work., IEEE, 2017: pp. 1132–1140.

[50] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, Y. Fu, "Image Super-Resolution Using Very Deep Residual Channel Attention Networks," in: 2018: pp. 294–310.

[51] H. Liu, T. Brailsford, Reproducing "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention," J. Phys. Conf. Ser. 2589 (2023) 012012.

[52] J. Redmon, A. Farhadi, "YOLOv3: An Incremental Improvement," (2018). http://arxiv.org/abs/1804.02767.

[53] S.T.A. Ali, A.H. Usama, I.R. Khan, M.M. Khan, A. Siddiq, "Mobile Registration Number Plate Recognition Using Artificial Intelligence," in: 2021 IEEE Int. Conf. Image Process., IEEE, 2021: pp. 944–948.

[54] C.-W. Yu, Y.-L. Chen, K.-F. Lee, C.-H. Chen, C.-Y. Hsiao, "Efficient Intelligent Automatic Image Annotation Method based on Machine Learning Techniques," in: 2019 IEEE Int. Conf. Consum. Electron. - Taiwan, IEEE, 2019: pp. 1–2.

[55] B. Zhang, X. Liu, C. Yue, S. Liu, X. Li, S.Y. Liang, L. Wang, "An imbalanced data learning approach for tool wear monitoring based on data augmentation," J. Intell. Manuf. (2023) 1-22.

[56] R. Yousef, G. Gupta, N. Yousef, M. Khari, "A holistic overview of deep learning approach in medical imaging," Multimed. Syst. 28 (2022) 881–914.

[57] W. Xiao, G. Kreiman, "Gradient-free activation maximization for identifying effective stimuli," arXiv preprint arXiv:1905.00378 (2019).

[58] Y. Gao, "News Video Classification Model Based on ResNet-2 and Transfer Learning," Secur. Commun. Networks. 2021 (2021) 1–9.

[59] S.-O. Shim, R. Imtiaz, A. Siddiq, I.R. Khan, "License Plates Detection and Recognition with Multi-Exposure Images," Int. J. Adv. Comput. Sci. Appl. 13 (2022).

[60] Your Home for Data Science, (2023).

[61] J. Anaya, A. Barbu, "RENOIR – A dataset for real low-light image noise reduction," J. Vis. Commun. Image Represent. 51 (2018) 144–154.

[62] D.P. Kingma, J. Ba, "Adam: A Method for Stochastic Optimization," arXiv preprint arXiv:1412.6980 (2014).

[63] C.-M. Fan, T.-J. Liu, K.-H. Liu, "SUNet: Swin Transformer UNet for Image Denoising," in: 2022 IEEE Int. Symp. Circuits Syst., IEEE, 2022: pp. 2333–2337.

[64] S. Anwar, N. Barnes, "Real Image Denoising With Feature Attention," in: 2019 IEEE/CVF Int. Conf. Comput. Vis., IEEE, 2019: pp. 3155–3164.

[65] J. Redmon, A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv preprint arXiv:1804.02767 (2018).