International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I4P136
Volume 72 Issue 4, 366-377, April 2024

Original Article

Convolutional Neural Network to Modify the Restoration of a CCTV E-Ticket Image

Christopher Alexander[1], Benfano Soewito[2]

[1,2]*Department of Computer Science, Bina Nusantara University, Jakarta, Indonesia.*

[1]*Corresponding Author : christopher.alexander@binus.ac.id*

Received: 14 November 2023 Revised: 20 February 2024 Accepted: 09 April 2024 Published: 24 April 2024

Abstract *- Traffic violations are now increasingly worrying the local community. There are many methods that can be used to minimize this incident, one of which is the government creating an E-Ticket program with CCTV to detect the number plates of vehicles that violate traffic. However, the resulting images from CCTV can cause difficulties for the authorities, and this is because the resolution of images produced from CCTV is not optimal; therefore, in this research, a program was created that uses the Convolutional Neural Network method with SwinIR and uses a Transformer. The dataset used is from ATCS Bandung. The data is in the form of a screenshot photo. The aim is to increase the resolution of the images taken from the CCTV. The final result of image restoration was 400%, and the percentage for recognizing police number plates was 90%. The percentage from the amount of clearly visible data/number of datasets x 100%.*

Keywords *- CCTV, Convolutional Neural Network, Deep Learning, E-Ticket, SwinIR, Image.*

1. Introduction

Traffic violations still often cause inconvenience to the environment and surrounding areas. One of the reasons for this is the lack of discipline of road users in obeying the traffic signs that have been implemented. In 2020, traffic violations committed by people around the Jakarta area reached 1,930,983 cases. With the data just mentioned, Polda Metro Jaya will reduce this figure during the implementation of Operation Patuh Jaya 2021. Raids, which used to be carried out frequently, now no longer have an impact or deterrent effect on violators, not only that, the impact of raids makes the streets congested and busy. From the data obtained, the use of CCTV in monitoring traffic conditions on the road is very necessary. Closed Circuit Television (CCTV) is the use of a camera with the function of capturing video images to transmit video signals to a specific location, in this case, tending to several sets of monitors.

Its function is different from television broadcasts, CCTV signals tend to be closed during transmission. The initial use of CCTV was to prevent crime. Current technological developments have made CCTV a tool used by authorities, for example the police, to carry out e-tickets. Even though CCTV cannot prevent crimes or criminal acts, it can help identify, control and observe a crime—situation at a particular location. To be able to use CCTV optimally, there are several features that need to be considered before using it for traffic. To make it easier to find out these features, a consultant is needed to explain the needs to the end user, so that they do not make the wrong purchase and can use it optimally. There are many incidents where the results do not match the detected image. This is because the resolution of the CCTV is not good. Therefore, an advanced method can be used to restore the resolution of images captured by CCTV cameras. With this, the police will be greatly helped in their work in observing CCTV so that they can avoid conflicts in the form of misunderstandings regarding the identification of violators. Along with the development of Artificial Intelligence, Machine Learning. Several techniques can be used to restore the resolution of an image. In the process of identifying important areas, AI can prioritize the allocation of computing resources to increase resolution in these areas.

There are several methods and theories used in video upscaling, first, there is the Convolutional Neural Network method. In the CNN algorithm, there will be several series of convolution and pooling operations that are carried out. With this operation, it will be easier to collect the detected features. The main special technique in the CNN algorithm is convolution, where a filter slides over the input and combines the input values plus the filter values in a feature map.

The final goal is for CNN to be able to recognize objects or images based on the features it detects. One of them is image upscaling. In image upscaling, there is a super-resolution technique. The main purpose of super-resolution is to produce sharper and more detailed images from low-resolution source images. This super-resolution technique also involves image processing with methods such as deep learning or detail recovery algorithms to create a higher-resolution version of the image. [1-4]

So, in this research, the Convolutional Neural Network (CNN) algorithm method will be applied to restore image resolution. First of all, you have to collect and prepare a data set in the form of images taken from CCTV E-Tilang. After detection, the dataset will be divided into two parts, namely the training set and the test set, and then the data set will be labels correctly. Second, image pre-processing will be carried out, namely adjusting the size of the images to match the input received by the CNN. Then, normalize the image to change the pixel values to the same scale to facilitate the training process.

Next there will be the creation of a CNN model, which will add convolution layers, max-pooling, and fully connected layers as needed. After the CNN model creation is complete, model training will be carried out to try to understand the patterns in the images, determining the loss function that the model will optimize during training. As training continues, the model will adjust its parameters to improve its ability to predict the CCTV license number. After the model has been trained and validated properly, the model will continue to predict license numbers for CCTV with low resolution quality. Then, the new desired results will appear.

2. Related Work

2.1. Detection of Vehicle Police Number Plates

A lot of research has been carried out related to vehicle number plate detection. Rizky Dwi Novyantika, in his research, used a Convolutional Neural Network (CNN) to detect vehicle number plates. However, this research is still unable to identify vehicle number plates properly. [25]

Research conducted by Donny Avianto used the momentum backpropagation neural network algorithm for vehicle number plate detection. This research is also still not enough to recognize the number plates of vehicles traveling in traffic at fairly high speeds. [5] The third study also carried out a combination of image processing and artificial condition networks to recognize vehicle number plates, but still could not determine and still made mistakes in reading the characters on vehicle plates. [6]

2.1.1. Quality Image Enhancement from Low-Resolution Camera

In this research, researchers performed image super-resolution to improve the quality of images taken by the DPED dataset. The method used is Convolutional Neural Network. The framework used is a Generative Adversarial Network, which has a Generator Network and a Discriminator Network. The training process is divided into 2, namely, the generalization stage and the validation stage.

In the generalization stage, images with low resolution will be input into the Generator Network to be converted into images with high resolution. Then, it enters the validation stage where the image that has been converted into a high-resolution image will be entered into the Discriminator Network to be tested for authenticity. The final result obtained from this research is that the PSNR and SSIM values increased from 20.46 and 0.9136 to 20.61 and 0.9189 when taken by the iPhone.

2.2. Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNN) is a development of Multilayer Perceptron (MLP) which is included in the feedforward type neural network. [26] This method was first introduced by Fukushima in 1998. This method also has many wide applications in recognition, face recognition and object detection, image characterization, and many more. CNN itself contains neurons where these neurons have weight, bias and activation function. There are several layers that these neurons have separately. There is an input layer, an output layer, and also multiple hidden layers, where the hidden layers contain convolutional layers, pooling layers, fully connected layers and several normalization layers. Convolutional layers apply convolution operations to combine two different sets of information—this copies feedback from individual neurons to visual stimuli. There are three main architectures combined by Convolutional Neural Networks, local receptive fields, shared weight, which is usually called filters, and spatial subsampling, which is usually called pooling. The words convolution are matrices whose job is to work on filters. [13] Convolution Neural Network (CNN) is an algorithm part of deep learning. CNN is included in the type of deep neural network because the network depth is high and is widely applied to image data. CNN is able to perform mathematical calculations on input consisting of many hidden layers. The hidden layer manages the input images and sends the processed results in the output section. CNN has a convolution layer, which is useful for extracting features and a classification part to classify features into classes determined during training. The training phase takes more time to complete and requires a capable computing device. [7-11] However, CNN's prediction process is quite fast and accurate. The convolutional neural network architecture network is presented in Figure 1.

2.2.1. Convolution Layer

Convolution layers are part of the CNN network architecture. At this stage, convolution operations are carried out on the image and filters to produce a new feature map. This convolution is the process that underlies the CNN network architecture. Convolution is a mathematical term where the application of a function to the output of another function repeatedly. The convolution operation applies the output function as a feature map from the input image. Input and output can be seen as two real-valued arguments. [40]

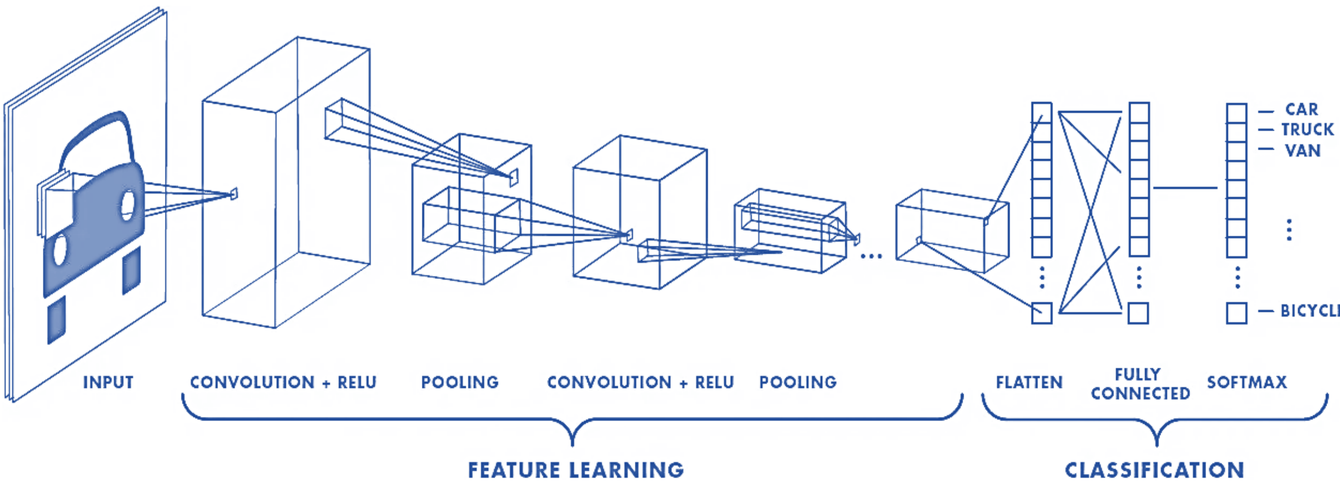


Fig. 1 The architecture of convolutional neural network

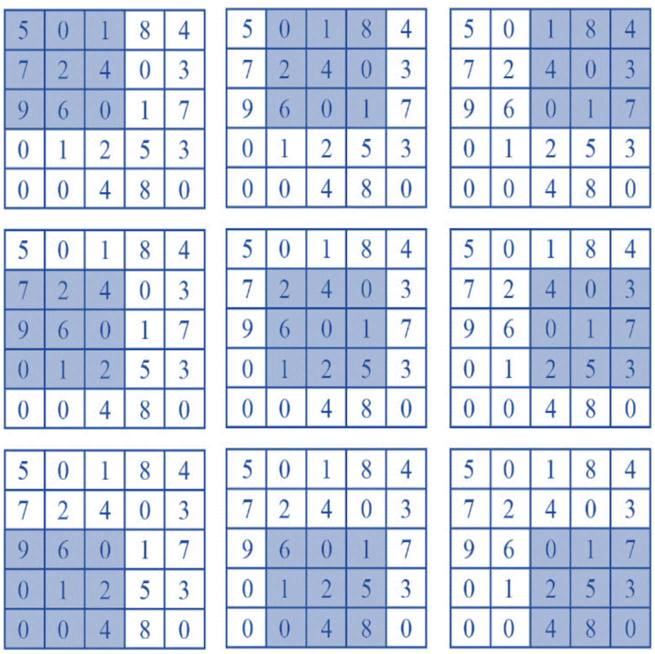


Fig. 2 Convolution Layer

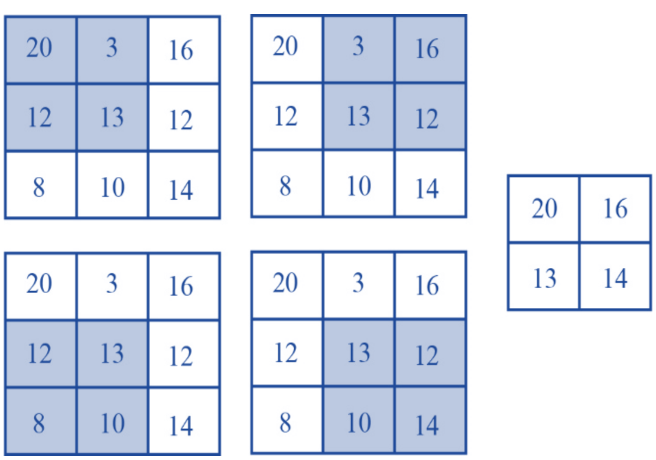


Fig. 3 & 4 Max Pooling Process & Result of Max Pooling

Fig. 5 Result of fully connected layer

2.2.2. Pooling Layer

Pooling layer is the process of reducing the size of image data. In image processing, the purpose of the pooling layer is to increase the position invariance of features. Then max pooling divides the output from the convolution layer into several small grids and then takes the maximum value from each grid to compile a reduced image matrix as shown in Figures 3 and 4. [40]

2.2.3. Fully Connected Layer

Fully connected layer, that is, the layer will take all the neurons in the previous layer (convolutional layer and max pooling layer) and connect them to every existing single neuron. This extensive connectivity enables the layer to fully process and synthesize the information from the entire previous layer. [15]

Fully connected layer is a layer that is usually used in implementing Multilayer Perceptron (MLP) and aims to carry out transformations on data dimensions so that data can be classified linearly. This transformation is crucial for tasks like classification, where the final goal is to linearly classify data points into different categories based on the learned features. This involves linear (and also non-linear through activation functions) transformations of the data dimensions as seen in Figure 5. [40]

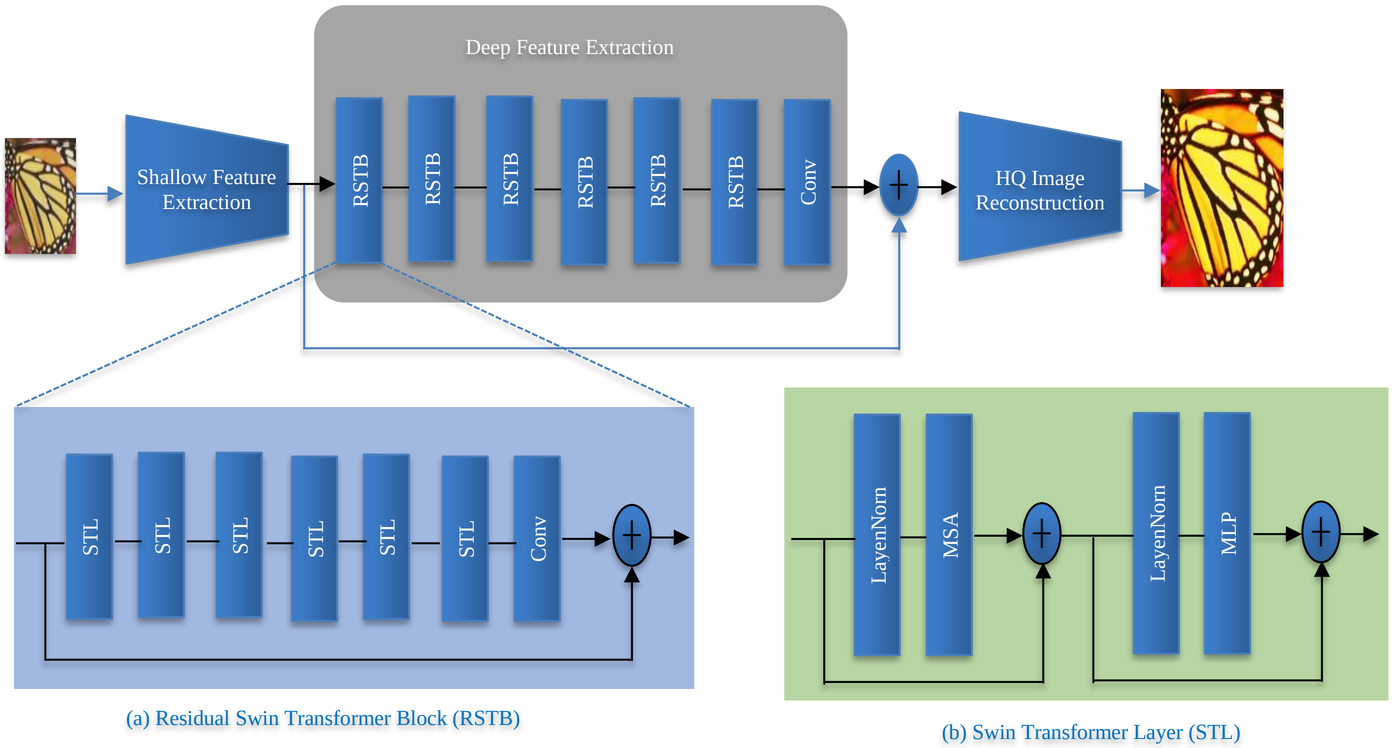


Fig. 6 The architecture of the proposed SwinIR for image restoration

3. Methods

3.1. Network Architecture

As shown in Figure 6, SwinIR consists of three modules: shallow feature extraction, deep feature extraction and highquality (*HQ*) image reconstruction modules. The researcher employs the same feature extraction modules for all restoration tasks but uses different reconstruction modules for different tasks. Shallow and deep feature extraction. Given a low-quality (*LQ*) $I_{LQ} \in \mathbb{R}^{HxWxC_{in}}$ (*H*, *W*, dan C_{in} are the image height, width and input channel number, respectively), the researcher uses a 3×3 convolutional layer $H_{SF}(\cdot)$ to extract shallow feature $F_0 \in \mathbb{R}^{HxWxC}$ as;

$$F_0 = H_{SF}(I_{LQ}),$$

Where *C* is the feature channel number. The convolution layer is good at early visual processing, leading to more stable optimization and better results. It also provides a simple way to map the input image space to a higher-dimensional feature space. Then, the extraction of deep features is executed. $F_{DF} \in \mathbb{R}^{HxWxC}$ *C* from F_0 as;

$$F_{DF} = H_{DF}(F_0),$$

Where $H_{DF}(\cdot)$ is the deep feature extraction module, and it contains K residual Swin Transformer blocks (RSTB) and a 3 × 3 convolutional layer. More specifically, intermediate features $F_1, F_2, \ldots, F_K$, and the output deep feature F_{DF} are extracted block by block as;

$$F_i = H_{RSTB_i}(F_{i-1}), i = 1, 2, \ldots, K,$$

$$F_{DF} = H_{CONV}(F_K),$$

Where $H_{RSTBI}(\cdot)$ denotes the *i*-th RSTB, and H_{CONV} is the last convolutional layer. Using convolutional layers at the end of feature extraction can bring the inductive bias of convolution operations into Transformer-based networks and lay a better foundation for later aggregation of Shallow and Deep Features.

3.2. Image Reconstruction

For instance, a reconstruction of the high-quality image I_{RHQ} by aggregating shallow and deep features using image SR.

$$I_{RHQ} = H_{REC}(F_0 + F_{DF}),$$

Taking $H_{REQ}(\cdot)$ as the function of module reconstruction. Shallow features generally have low frequencies, while deep features focus on recovering high frequencies that have been lost. With a long connection pass, SwinIR can transmit information from low frequencies and stabilize training. Reconstruction module, sub-pixel ¬convolution layer is used to enhance the features for the implementation.[43, 44]

Reconstruction using a single convolution layer is utilized for tasks like image denoising and JPEG compression artifact reduction that do not require up-sampling. Furthermore, rather than reconstructing the HQ image, the researcher applies residual learning to reconstruct the residual between the LQ and HQ images. It is constructed as

$$I_{RHQ} = H_{SwinIR}(I_{LQ}) + I_{LQ},$$

Where $H_{SwinIR}(\cdot)$ shows the function of SwinIR.

3.3. Loss Function

Researchers minimize L1 pixel loss to optimize the SwinIR settings for SR pictures.

$$\mathcal{L} = \|I_{RHQ} + I_{HQ}\|_1,$$

Where I_{RHQ} is the equivalent ground truth HQ image and IRHQ is produced by using ILQ as input from SwinIR. To illustrate the efficacy of the suggested network, we solely employ naïve L1 pixel loss for traditional and lightweight picture SR, as in earlier research. To enhance visual quality for real-world SR photos, pixel loss, GAN loss, and perception loss are combined. [16, 41, 12, 14, 42] Researchers employ Charbonnier Loss to reduce artifacts from JPEG compression and image noise. For image noise reduction and JPEG compression artifact reduction, the researcher uses Charbonnier Loss. [17 - 25]

$$\mathcal{L} = \sqrt{\|I_{RHQ} + I_{HQ}\|^2 + \epsilon^2},$$

Where ϵ is a constant empirically set to 10^{-3}.

4. Experiment

4.1. CNN Experiment

In this test, the researcher applied several stages in order to carry out image restoration using the CNN method, namely:

- Step 1: Upload Dataset
- Step 2: Input layer
- Step 3: Convolutional layer
- Step 4: Pooling layer
- Step 5: Second Convolutional Layer and Pooling Layer
- Step 6: Dense layer
- Step 7: Logit Layer

The combination of functions of the stages in CNN and when used in testing and evaluating models is as follows.

```
def cnn_model_fn(features, labels, mode):
  """Model function for CNN."""
# Input Layer
  input_layer = tf.reshape(features["x"], [-
1, 28, 28, 1])

# Convolutional Layer
  conv1 = tf.layers.conv2d(
      inputs=input_layer,
      filters=32,
      kernel_size=[5, 5],
      padding="same",
      activation=tf.nn.relu)
# Pooling Layer
  pool1 =
tf.layers.max_pooling2d(inputs=conv1,
pool_size=[2, 2], strides=2)
# Convolutional Layer #2 and Pooling Layer
  conv2 = tf.layers.conv2d(
      inputs=pool1,
      filters=36,
      kernel_size=[5, 5],
      padding="same",
      activation=tf.nn.relu)
  pool2 =
tf.layers.max_pooling2d(inputs=conv2,
pool_size=[2, 2], strides=2)

# Dense Layer
  pool2_flat = tf.reshape(pool2, [-1, 7 * 7
* 36])
  dense = tf.layers.dense(inputs=pool2_flat,
units=7 * 7 * 36, activation=tf.nn.relu)
  dropout = tf.layers.dropout(
      inputs=dense, rate=0.4, training=mode
== tf.estimator.ModeKeys.TRAIN

# Logits Layer
  logits = tf.layers.dense(inputs=dropout,
units=10)

  predictions = {
# Generate predictions (for PREDICT and EVAL
mode)
      "classes": tf.argmax(input=logits,
axis=1),
      "probabilities": tf.nn.softmax(logits,
name="softmax_tensor")
  }

  if mode == tf.estimator.ModeKeys.PREDICT:
    return
tf.estimator.EstimatorSpec(mode=mode,
predictions=predictions)

# Calculate Loss
  loss =
tf.losses.sparse_softmax_cross_entropy(label
s=labels, logits=logits)

# Configure the Training Op (for TRAIN mode)
  if mode == tf.estimator.ModeKeys.TRAIN:
    optimizer =
tf.train.GradientDescentOptimizer(learning_r
ate=0.001)
    train_op = optimizer.minimize(
        loss=loss,

global_step=tf.train.get_global_step())
    return
tf.estimator.EstimatorSpec(mode=mode,
loss=loss, train_op=train_op)

# Add evaluation metrics Evaluation mode
  eval_metric_ops = {
      "accuracy": tf.metrics.accuracy(
```

```
          labels=labels,
predictions=predictions["classes"])}
  return tf.estimator.EstimatorSpec(
      mode=mode, loss=loss,
eval_metric_ops=eval_metric_ops)
```

4.2. SwinIR Experiment

In carrying out photo restoration, researchers need materials to test the application, researchers call this the training stage. In the training process, researchers carried out several material settings, which are described in the following source code:

(setting1: training model size = 48 pixel)

```
python main_test_swinir.py --task
classical_sr --scale 2 --training_patch_size
48 --model_path
model_zoo/swinir/001_classicalSR_DIV2K_s48w8
_SwinIR-M_x2.pth --folder_lq
testsets/Set5/LR_bicubic/X2 --folder_gt
testsets/Set5/HR
python main_test_swinir.py --task
classical_sr --scale 3 --training_patch_size
48 --model_path
model_zoo/swinir/001_classicalSR_DIV2K_s48w8
_SwinIR-M_x3.pth --folder_lq
testsets/Set5/LR_bicubic/X3 --folder_gt
testsets/Set5/HR
python main_test_swinir.py --task
classical_sr --scale 4 --training_patch_size
48 --model_path
model_zoo/swinir/001_classicalSR_DIV2K_s48w8
_SwinIR-M_x4.pth --folder_lq
testsets/Set5/LR_bicubic/X4 --folder_gt
testsets/Set5/HR
python main_test_swinir.py --task
classical_sr --scale 8 --training_patch_size
48 --model_path
model_zoo/swinir/001_classicalSR_DIV2K_s48w8
_SwinIR-M_x8.pth --folder_lq
testsets/Set5/LR_bicubic/X8 --folder_gt
testsets/Set5/HR
```

(setting2: training model size = 64 pixel)

```
python main_test_swinir.py --task
classical_sr --scale 2 --training_patch_size
64 --model_path
model_zoo/swinir/001_classicalSR_DF2K_s64w8_
SwinIR-M_x2.pth --folder_lq
testsets/Set5/LR_bicubic/X2 --folder_gt
testsets/Set5/HR
python main_test_swinir.py --task
classical_sr --scale 3 --training_patch_size
64 --model_path
model_zoo/swinir/001_classicalSR_DF2K_s64w8_
SwinIR-M_x3.pth --folder_lq
testsets/Set5/LR_bicubic/X3 --folder_gt
testsets/Set5/HR
python main_test_swinir.py --task
classical_sr --scale 4 --training_patch_size
64 --model_path
model_zoo/swinir/001_classicalSR_DF2K_s64w8_
SwinIR-M_x4.pth --folder_lq
testsets/Set5/LR_bicubic/X4 --folder_gt
testsets/Set5/HR
python main_test_swinir.py --task
classical_sr --scale 8 --training_patch_size
64 --model_path
model_zoo/swinir/001_classicalSR_DF2K_s64w8_
SwinIR-M_x8.pth --folder_lq
testsets/Set5/LR_bicubic/X8 --folder_gt
testsets/Set5/HR
```

(setting 3: training model with lesslight and size = 128 pixels)

```
python main_test_swinir.py --task
lightweight_sr --scale 2 --model_path
model_zoo/swinir/002_lightweightSR_DIV2K_s64
w8_SwinIR-S_x2.pth --folder_lq
testsets/Set5/LR_bicubic/X2 --folder_gt
testsets/Set5/HR
python main_test_swinir.py --task
lightweight_sr --scale 3 --model_path
model_zoo/swinir/002_lightweightSR_DIV2K_s64
w8_SwinIR-S_x3.pth --folder_lq
testsets/Set5/LR_bicubic/X3 --folder_gt
testsets/Set5/HR
python main_test_swinir.py --task
lightweight_sr --scale 4 --model_path
model_zoo/swinir/002_lightweightSR_DIV2K_s64
w8_SwinIR-S_x4.pth --folder_lq
testsets/Set5/LR_bicubic/X4 --folder_gt
testsets/Set5/HR
```

(setting 4: training model grayscale dan middle size 128)

```
python main_test_swinir.py --task gray_dn --
noise 15 --model_path
model_zoo/swinir/004_grayDN_DFWB_s128w8_Swin
IR-M_noise15.pth --folder_gt testsets/Set12
python main_test_swinir.py --task gray_dn --
noise 25 --model_path
model_zoo/swinir/004_grayDN_DFWB_s128w8_Swin
IR-M_noise25.pth --folder_gt testsets/Set12
python main_test_swinir.py --task gray_dn --
noise 50 --model_path
model_zoo/swinir/004_grayDN_DFWB_s128w8_Swin
IR-M_noise50.pth --folder_gt testsets/Set12
```

(setting 5: training model colorful and middle size 128)

```
python main_test_swinir.py --task color_dn -
-noise 15 --model_path
model_zoo/swinir/005_colorDN_DFWB_s128w8_Swi
nIR-M_noise15.pth --folder_gt
testsets/McMaster
python main_test_swinir.py --task color_dn -
-noise 25 --model_path
model_zoo/swinir/005_colorDN_DFWB_s128w8_Swi
nIR-M_noise25.pth --folder_gt
testsets/McMaster
python main_test_swinir.py --task color_dn -
-noise 50 --model_path
model_zoo/swinir/005_colorDN_DFWB_s128w8_Swi
nIR-M_noise50.pth --folder_gt
testsets/McMaster
```

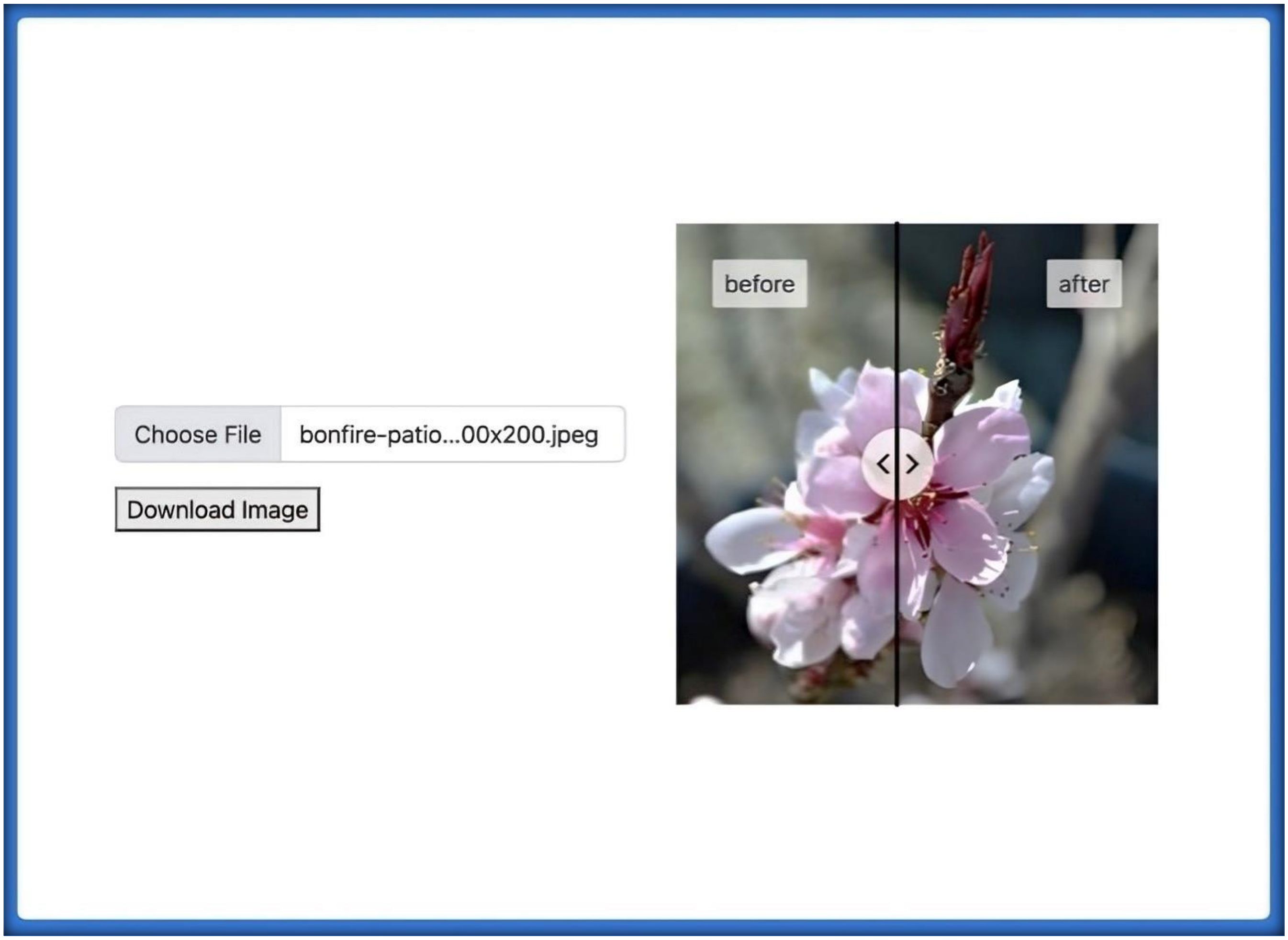


Fig. 7 Photo restoration using training data.

From the results of the training data, all of them produce a good increase in resolution to be able to clarify the photos. It can be seen from one of the photo resolution results of the training data, the before and after results in Figure 7 above.

Then, this will be applied to the dataset that has been collected by researchers to obtain photo restoration results using the CNN (SwinIR) method.

4.3. Photo Restoration Results using the CNN method

By using an application that has been designed using the CNN method, the difference can be seen between before the restoration was carried out and after the restoration was carried out. The image becomes clearer and sharper.

In Figure 7 above, you can see how the application restores photo data smoothly, and the results are recorded in the table 1.

Based on the results of photo restoration, test data that has been carried out on the dataset using the convolutional neural network method, a percentage increase in resolution can be obtained using the formula below.

$$resolution\ percentage = \frac{result\ resolution}{initial\ resolution}\ x\ 100\%$$

$$resolution\ percentage = \frac{604}{151}\ x\ 100\%$$

$$resolution\ percentage = 400\%$$

(Taken from sampling data in dataset 1)

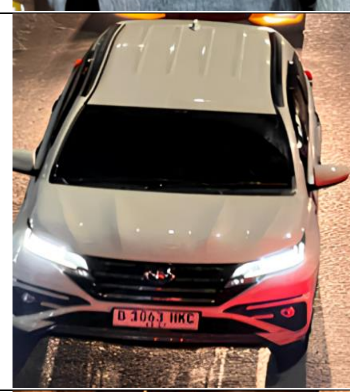
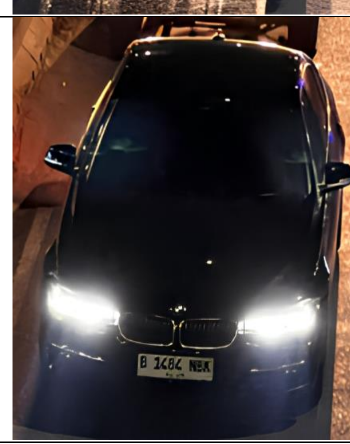
Moreover, the percentage, which is based on whether the police number is clearly visible or not, is obtained using the formula below.

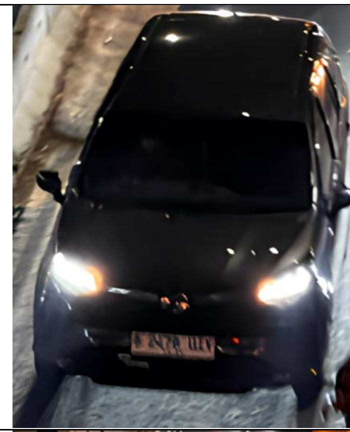
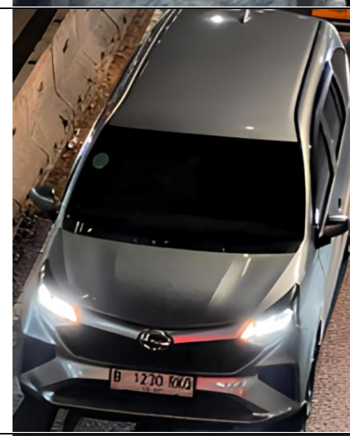
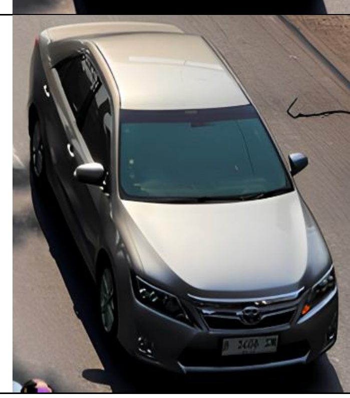
$$Clarity\ Percentage = \frac{Sum\ of\ Data\ Visible}{Number\ of\ Datasets}\ x\ 100\%$$

$$Clarity\ Percentage = \frac{9}{10}\ x\ 100\%$$

$$Clarity\ Percentage = 90\%$$

Table 1. Photo restoration test results

No	Data Name	Data resolution	Data Display	Data Results Resolution	Result Display	Quality Results
1.	Screenshot_1	(151 x 203) pixels		(604 x 812) pixels		Police Number is Clearly Visible.
2.	Screenshot_2	(201 x 204) pixels		(804 x 816) pixels		Police Number is Clearly Visible.
3.	Screenshot_3	(171 x 251) pixels		(684 x 1004) pixels		Police Number is Clearly Visible.
4.	Screenshot_4	(176 x 191) pixels		(704 x 764) pixels		Police Number is Clearly Visible.

5.	Screenshot_5	(138 x 181) pixels		(552 x 724) pixels		The police number is not yet clearly visible.
6.	Screenshot_6	(150 x 195) pixels		(600 x 780) pixels		Police Number is Clearly Visible.
7.	Screenshot_7	(144 x 168) pixels		(576 x 672) pixels		Police Number is Clearly Visible.
8.	Screenshot_8	(184 x 181) pixels		(736 x 724) pixels		Police Number is Clearly Visible.

9.	Screenshot_9	(135 x 173) pixels		(540 x 692) pixels		Police Number is Clearly Visible.
10.	Screenshot_10	(145 x 181) pixels		(580 x 724) pixels		Police Number is Clearly Visible.

5. Conclusion and Recommendation

5.1. Conclusion

Based on the results of the research conducted, it can be concluded that the implementation of photo restoration requires the CNN method to produce maximum photo restoration and make photos clear. When implementing traffic rules, especially in odd-even areas, using the CNN reversion method, the police can detect the number of passing vehicles so that they can use evidence of traffic violations to reduce traffic jams. The increase in resolution from the photo restoration results was obtained by 400%, where with this increase the percentage of recognizing police number plates was 90%. [27-39]

5.2. Recommendation

The police have not carried out the implementation of photo restoration as law enforcers. It is hoped that this research can be used as a reference by law enforcers to be able to apply technology to improve photo restoration using the CNN method as a law enforcement tool.

References

[1] Anamika Dhillon, and Gyanendra K. Verma, “Convolutional Neural Network: A Review of Models, Methodologies and Applications to Object Detection,” *Progress in Artificial Intelligence*, vol. 9, pp. 85-112, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[2] Augyeris Lioga Seandrio, Awang Hendrianto Pratomo, and Mangaras Yanu Florestiyanto, “Implementation of Convolutional Neural Network (CNN) in Facial Expression Recognition,” *Telematics: Journal of Informatics and Information Technology*, vol. 18, no. 2, pp. 211-221, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[3] Boki Latupono, “*Implementation of Deep Learning Using Convolution Neural Network for Image Classification*,” Department of Statistics, Faculty of Mathematics and Natural Sciences Indonesian Islamic University, 2018. [Google Scholar] [Publisher Link]

[4] Sisilia Daeng Bakka Mau, “The Effect of Histogram Equalization for Improving Digital Image Quality,” *Symmetric: Journal of Mechanical, Electrical and Computer Science Engineering*, vol. 7, no. 1, pp. 117-182, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[5] Donny Avianto, “Recognition of Vehicle Number Plate Character Patterns Using the Momentum Backpropagation Neural Network Algorithm,” *Journal of Informatics*, vol. 10, no. 1, pp. 1199-1209, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[6] Helmy Fitriawan, Ouriz Pucu, and Yohanes Baptista, “Off-Line Vehicle Number Plate Identification Based on Image Processing and

Artificial Neural Networks," *ELECTRICIAN – Journal of Electrical Engineering and Technology,* vol. 6, no. 2, pp. 123-126, 2012. [Google Scholar] [Publisher Link]

[7] Firoz Mahmud et al., "Face Recognition using Principle Component Analysis and Linear Discriminant Analysis," *International Conference on Electrical Engineering and Information Communication Technology*, Savar, Bangladesh, pp. 1-4, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[8] Faiq Nukha, "*Classification of CCTV Camera Determination Based on Road Type Using Deep Neural Network Algorithm (DNN)*," Undergraduate Thesis, Universitas Islam Negeri Maulana Malik Ibrahim, pp. 1-105, 2019. [Google Scholar] [Publisher Link]

[9] Fahima Tabassum et al., "Human Face Recognition with Combination of DWT and Machine Learning," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 3, pp. 546-556, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] Halprin Abhirawa, Jondri Jondri, and Anditya Arifianto, "Face Recognition Using Convolutional Neural Network," *eProceedings of Engineering*, vol. 4, no. 3, pp. 4907-4916, 2017. [Google Scholar] [Publisher Link]

[11] Haoyu Xu et al., "Foreign Object Debris Material Recognition Based on Convolutional Neural Networks," *EURASIP Journal on Image and Video Processing*, vol. 2018, pp. 1-10, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[12] Ian Goodfellow et al., "Generative Adversarial Nets," *Advances in Neural Information Processing Systems*, pp. 2672-2680, 2014. [Google Scholar] [Publisher Link]

[13] Jamie Ludwig, "*Image Convolution*," pp. 1-8, 2012. [Google Scholar] [Publisher Link]

[14] Justin Johnson, Alexandre Alahi, and Li Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution," *European Conference on Computer Vision*, Springer, vol. 9906, pp. 674-711, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[15] Junwei Han et al., "Advanced Deep-Learning Techniques for Salient and Category-Specific Object Detection: A Survey," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 84-100, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[16] Kai Zhang et al., "Designing a Practical Degradation Model for Deep Blind Image Super-Resolution," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, pp. 4791-4800, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[17] Kamal Hasan Mahmud, Adiwijaya Adiwijaya, and Said Al Faraby, "Multi-class Image Classification Using Convolutional Neural Network," *eProceedings of Engineering*, vol. 6, no. 1, pp. 2127-2136, 2019. [Google Scholar] [Publisher Link]

[18] Li Deng, and Dong Yu, "Deep Learning: Methods and Applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3-4, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[19] M. Shamim Hossain, and Ghulam Muhammad, "Emotion Recognition Using Deep Learning Approach from Audio–Visual Emotional Big Data," *Information Fusion*, vol. 49, pp. 69-78, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[20] M. Hema Latha, S. Varadarajan, and Y. Murali Mohan Babu, "Comparison of DWT, DWT-SWT, and DT-CWT for Low Resolution Satellite Images Enhancement in Emerging," *International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies*, Chennai, India, pp. 1-5, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[21] Md Zahangir Alom et al., "The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches," *arXiv*, pp. 1-39, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[22] Mohammad Sadegh Norouzzadeh et al., "Automatically Identifying, Counting, and Describing Wild Animals in Camera-Trap Images With Deep Learning," *Biological Sciences*, vol. 115, no. 25, pp. E5716-E5725, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[23] Mohammadreza Babaee, Duc Tung Dinh, and Gerhard Rigoll, "A Deep Convolutional Neural Network for Video Sequence Background Subtraction," *Pattern Recognition*, vol. 76, pp. 635-649, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[24] Nopita Pratiwi Patmawati, "*Improving Image Quality from Low Resolution Cameras using Convolutional Neural Network*," Telkom University, S1 Informatics, 2020. [Publisher Link]

[25] Pierre Charbonnier et al., "Two Deterministic Half-Quadratic Regularization Algorithms for Computed Imaging," *International Conference on Image Processing*, vol. 2, pp. 168-172, 1994. [CrossRef] [Google Scholar] [Publisher Link]

[26] Pulung Adi Nugroho, Indah Fenriana, and Rudy Arijanto, "Implementation of Deep Learning using Convolutional Neural Network (CNN) in Human Expressions," *Algor*, vol. 2, no. 1, pp. 1-10, 2020. [Google Scholar] [Publisher Link]

[27] Rajeev Ranjan et al., "Deep Learning for Understanding Faces: Machines May Be Just as Good, or Better, than Humans," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 66-83, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[28] Rizky Dwi Novyantika, "*Detection of Motorized Vehicle Number Signs on Streaming Media Using the Convolutional Neural Network Algorithm Using Tensorflow*," Undergraduate Thesis, Faculty of Mathematics and Natural Sciences, 2018. [Google Scholar] [Publisher Link]

[29] Roberto Olmos, Siham Tabica, and Francisco Herrera, "Automatic Handgun Detection Alarm in Videos using Deep Learning," *Neurocomputing*, vol. 275, pp. 66-72, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[30] Rui Zhao et al., "Deep Learning and Its Applications to Machine Health Monitoring," *Mechanical Systems and Signal Processing*, vol. 115, pp. 213-237, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[31] Ari Peryanto, Anton Yudhana, and dan Rusydi Umar, "Image Classification Design Using Deep Learning Technology Based on Convolutional Neural Network Methods," *Format Journal*, vol. 8, no. 2, pp. 138-147, 2019. [Google Scholar]

[32] S. Ravi, and Sadique Nayeem, "A Study on Face Recognition Technique based on Eigenface," *International Journal of Applied Information Systems*, vol. 5, no. 4, pp. 57-62, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[33] Shiv Gehlot, Anubha Gupta, and Ritu Gupta, "SDCT-AuxNetθ: DCT Augmented Stain Deconvolutional CNN with Auxiliary Classifier for Cancer Diagnosis," *Medical Image Analysis*, vol. 61, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[34] S. Suja, Nimmy George, and Annie George, "Classification of Grades of Astrocytoma Images from MRI Using Deep Neural Network," *2nd International Conference on Trends in Electronics and Informatics*, Tirunelveli, India, pp. 1257-1262, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[35] Sung Cheol Park, Min Kyu Park, and Moon Gi Kang, "Super-Resolution Image Reconstruction: A Technical Overview," *IEEE Signal Processing Magazine*, vol. 20, pp. 21-36, 2003. [CrossRef] [Google Scholar] [Publisher Link]

[36] Syarifah Rosita Dewi, "*Deep Learning Object Detection in Video Using Tensorflow and Convolutional Neural Network*," Undergraduate Thesis, Faculty of Mathematics and Natural Sciences, 2018. [Google Scholar] [Publisher Link]

[37] Titik Khotiah et al., "Automatic Vehicle Number Plate Detection Using Convolutional Neural Network and OCR at ITB Ahmad Dahlan Lamongan Parking LoT," *Journal of Information Management & Information Systems*, vol. 6, no. 2, pp. 114-122, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[38] Wailan Thom Tirajoh, "*One-Shot Learning Face Recognition for Academic Attendance using Deep Convolutional Neural Network*," S1 Thesis, Universitas Atma Jaya Yogyakarta, 2020. [Google Scholar] [Publisher Link]

[39] Vandel Maha Putra Salawazo et al., "Implementation of the Convolutional Neural Network (CNN) Method in CCTV Video Object Recognition," *Mantik Penusa Journal*, vol. 3, no. 1.1, pp. 74-79, 2019. [Google Scholar] [Publisher Link]

[40] Wayan Suartika Eka Putra, "Image Classification Using Convolutional Neural Network (CNN) at Caltech 101," *ITS Engineering Journal*, vol. 5, no. 1, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[41] Hendry Wiranto, "*Facial Expression Recognition Using Wavelet Transform and Convolutional Neural Network*," Thesis, pp. 1-129, 2019. [Google Scholar] [Publisher Link]

[42] Xiao Ning, Wen Zhu, and Shifeng Chen, "Recognition Object Detection, and Segmentation of White Background Photos Based on Deep Learning," *32nd Youth Academic Annual Conference of Chinese Association of Automation*, Hefei, China, pp. 182-187, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[43] Xintao Wang et al., "Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data," *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Montreal, BC, Canada, pp. 1905-1914, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[44] Xintao Wang et al., "ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks," *European Conference on Computer Vision Workshops*, pp. 701-710, 2018. [CrossRef] [Google Scholar] [Publisher Link]