



metric [11].

## II. LITERATURE REVIEW

### A. Theoretical Foundation

#### 1) Super resolution

The image super resolution, also known as Single-Image Super Resolution (SISR), refers to a computer vision task that aims to recover or restore a low-resolution image into high-resolution, by performing spatial resolution enhancement [12]. In recent years, deep learning-based SISR models have been actively explored. The recent advances leverage deep learning, particularly in Convolutional Neural Network (CNN) and Generative Adversarial Networks (GANs) [13–15].

Up sampling an image represents one of several straightforward solutions for enhancing low-resolution images. However, applying raw upscaling directly to a low-resolution image can lead to image artifacts, including blurriness or noise. Consequently, in the context of deep learning-based super-resolution tasks, a more effective approach involves utilizing learnable up-sampling layers well-known as Subpixel. These specialized layers are directly derived from convolutional layers. By incorporating learnable up-sampling, the model acquires the capability to perform super-resolution tasks [12].

To evaluate a SISR performance, the common approach is by comparing the enhanced low-resolution image from its high-resolution counterpart. Some objective metrics, like Mean Squared Error (MSE), Structural Similarity Index Measurement (SSIM), and Peak Signal-to-Noise Ratio (PSNR) [16, 17], each had their own accounting factors to be evaluated. The MSE mainly compares color composition differential distance between the predicted and the ground truth (Eq. (1)) [18]. SSIM counts the color composition based-on covariance and means which resembles the contrast and luminance-related factors (Eq. (2)) [18]. This makes SSIM great for accounts, the fixed main structure. PSNR, an MSE-based formula (Eq. (3)) accounts the maximum difference between pixel values error of image reconstruction (Eq. (4)) [19]. As super resolution task endeavor enhancing the low-resolution image to match its high-resolution counterpart, performing scoring through three previously mentioned methods (MSE, SSIM, and PSNR) for quality control is unreliable. This is because these three metrics are primarily focused on pixel-to-pixel comparisons, while the true objective is having fixed or generated image that perceptually indistinguishable from the high-resolution counterpart. Ledig *et al.* introduced a deep learning specific metric called perceptual loss [13]. The term “loss” came from its specific role to perform loss scoring system, which is suitable for minimization problem in deep learning backpropagation process during training. Perceptual loss objectively measures perceptual difference between both generated and high-resolution images utilizing pretrained Visual Geometry Group 19 (VGG19) model feature extractions process [13, 16]. Both feature extraction result, then had their differential measured using L2 norm, which in this case, was MSE. The content loss (Eq. (5)) then completed by adding it with adversarial loss (Eq. (6)) [13, 16].

#### a) Mean squared error

$$MSE(x, y) = \frac{1}{N} \cdot \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} [x(i, j) - y(i, j)]^2 \quad (1)$$

where:

$N$  stands for total data.

$x$  stands for generated image.

$y$  stands for high-resolution image or ground truth.

$w$  and  $h$  stand for width and height respectively.

$i$  and  $j$  stand for iterated width and height respectively.

#### b) Structural similarity index measurement formula

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)}{(\mu_x^2 + \mu_y^2 + c_1)} \cdot \frac{(2\sigma_{xy} + c_2)}{(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (2)$$

where:

$x$  stands for generated image.

$y$  stands for high-resolution image or ground truth.

$\mu_x$  and  $\mu_y$  stand for mean intensities of  $x$  and  $y$  respectively. The mu mainly resembles luminance-related variables [18].

$\sigma_x$  and  $\sigma_y$  stands for standard deviations of pixel intensities in  $x$  and  $y$ , respectively. The sigma mainly resembles contrast-related variables [18].

$c$  stands for coefficient to prevent any division by zero.

#### c) Mean squared error for PSNR formula

$$MSE(x, y) = \frac{1}{w \cdot h} \cdot \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} [x(i, j) - y(i, j)]^2 \quad (3)$$

where:

$x$  stands for generated image.

$y$  stands for high-resolution image or ground truth.

$w$  and  $h$  stand for width and height respectively.

$i$  and  $j$  stand for iterated width and height respectively.

#### d) Peak signal-to-noise ratio formula

$$PSNR(x, y) = 20 \cdot \log_{10} \left( \frac{255^2}{\sqrt{MSE(x, y)}} \right) \quad (4)$$

where:

$x$  stands for generated image.

$y$  stands for high-resolution image or ground truth.

$MSE$  refers to Eq. (3) since PSNR uses MSE [19].

#### e) Content loss formula in perceptual loss

$$content_{loss}(I^{LR}, I^{HR}) = \frac{1}{w_{i,j} \cdot h_{i,j}} \cdot \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} \left( \phi_{i,j} \cdot I_{x,y}^{HR} - \phi_{i,j} \cdot (G_{\theta} \cdot (I^{LR})_{x,y}) \right)^2 \quad (5)$$

where:

$N$ ,  $n$ ,  $i$ , and  $j$  stand for dataset  $n$ -th.

$I^{LR}$  stands for low-resolution image.

$I^{HR}$  stands for high-resolution image or ground truth.

$G$  stands for the generator model.

$\Theta$  stands for tensor. If denoted with model expression, it refers to produced model tensor.

$x$  and  $y$  represent image pixel-th, stands for width and height respectively.

$\Phi$  stands for VGG-19 loss.

#### f) Perceptual Loss formula

$$loss(I^{LR}, I^{HR}) = content_{loss} + \sum_{n=1}^N -\log D_{\theta}(G_{\theta}(I^{LR})) \quad (6)$$

where:

$Content_{loss}$  refers to Eq. (5).

$N$  and  $n$  stand for dataset  $n$ -th.

$I^{LR}$  stands for low-resolution image.

$I^{HR}$  stands for high-resolution image or ground truth.

$D$  stands for the discriminator model.

$G$  stands for the generator model.

$\Theta$  stands for tensor. If denoted with model expression, it refers to produced model tensor.

## 2) Deep learning

Deep learning is a subset of machine learning that employs artificial neural networks (ANN) architecture as shown in Fig. 1, built-in to analyze complex patterns and relationship in data [20]. This advancement in machine learning, mainly inspired by the human neural process, which employs the massive numbers of brain neurons and learning things iteratively.

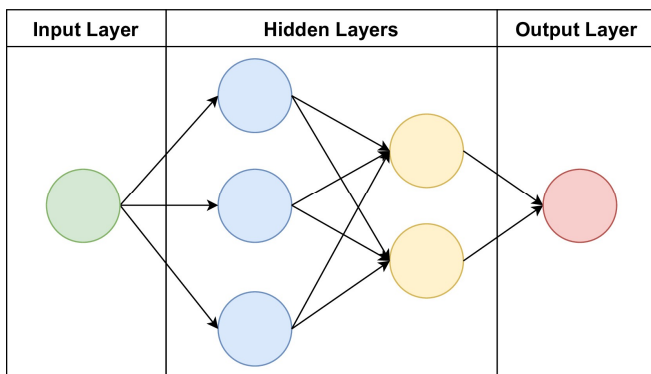


Fig. 1. Artificial neural network architecture.

As an ANN, this model consists of input, hidden, and output layers. Every time data goes through a layer, leveraging its neurons, the layer gives weight for each neuron or kernel to figure out what should it done. In general, the input layer employed as the front gate for to be learned data. The hidden layers consist of one or multiple layers that the main purpose is to learn the data. Last is the output layer, which is employed to give output for the model. What distinguishes ANN architecture from regular machine learning is all layer behaviors are customizable in-term of initialization, feedback through activation function, and regularization [20].

## 3) Convolutional neural network

In the pre-Convolutional Neural Network (CNN) era, performing a predictive computer vision task, the common approach was to perform two distinctives tasks. First, to perform image feature extraction or selection was necessary because raw image cannot be directly processed into a computer. Second, to perform predictive analysis such as classification and regression analysis. For example, when employing technology like Artificial Neuro-Fuzzy Inference System (ANFIS) [21].

This is where CNN takes a role. As a subset of deep learning model that mainly aims to process image, it completely unites the two distinctives process into single pipeline, where the convolutional layer works as feature extractor, and the rest hidden and output layer work as predictive layer [22].

Intuitively, the primary distinction of convolutional layer from linear layer is its tensor matrix shape. Unlike the linear layer, that primarily contains only neurons for performing feature extraction, the convolutional layer is a multi-

dimensional space. This enables the layer to perform feature extraction in a wider range. This is part where the CNN was originally targeted for processing image. Essentially, the image is a matrix of collection of numbers, that forms a meaningful data that resembles the spatial position, denoted as height and width, with color information denoted as single (black-white images) or three (red-green-blue) channels. This opens higher capabilities on learning more complex use cases, such as working with image which considered as a multi-dimensional data. Fig. 2 shows an example of how a convolutional layer works with kernel size of  $3 \times 3$ , with filters size of 16, hence the whole image is sized at  $6 \times 6$ . Each kernel (represented as a box that contains 16 circles in Fig. 2) contains learning weights, referred as “filters” (represented as circles within kernel in Fig. 2), which extracting information by learning specific area of the image to perform image analysis.

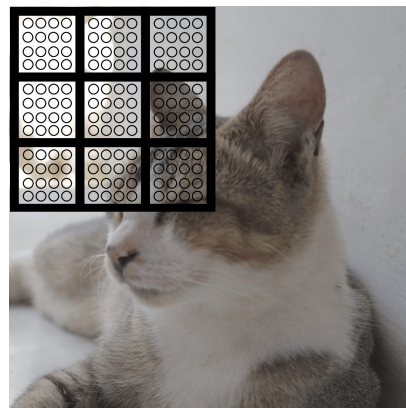


Fig. 2. The intuitive explanation image that describes how convolutional layer looks like when performing image feature extraction.

## 4) Generative adversarial network

Generative Adversarial Networks (GANs) are a subset of deep learning model that is classified as generative AI. The GANs consists of a generator and discriminator model. The generator model is used for performing generative tasks, such as generating image, style transfer, and super resolution. And the discriminator model is used for training the generator model by performing binary image classification, that can detect any image that is generated by the generator model [9]. The distinction arises from the nature of generative tasks, which primarily involve in creative creations.

GANs stands out in its adversarial training technique. When training GANs, both generator and discriminator models are trained [9]. But the common technique to perform an adversarial training, is to add the loss of generator model with the output of discriminator prediction as bias. The discriminator prediction output is referred to as discriminator loss as shown in Fig. 3. With AI expected to mimic human behavior, the generator model must achieve its mastery in producing output that are manmade. Because in the beginning of training the generator model, the output did not immediately intuitively look real in human perspective. This answer the basic question of why human intuitive still able to distinctively decide whether any creative creation is AI generated. Therefore, the goal of the adversarial training technique is to build a generator model that can fool the discriminator model, while also training the discriminator model to determine which outputs are fake/generated or real

patches, which the condition stated as *nash equilibrium*, as two models compete against each other [23].

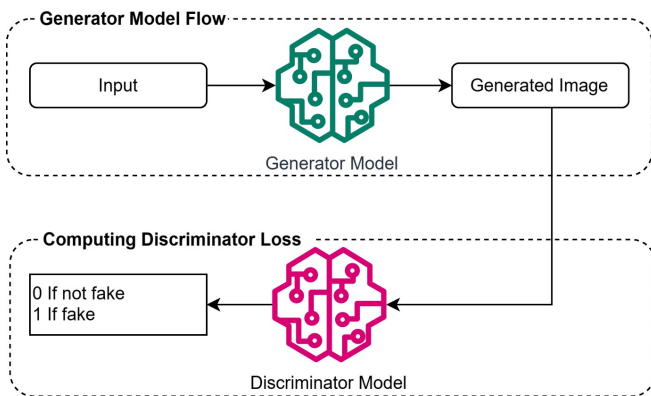


Fig. 3. Adversarial training flow, employing discriminator model to compute discriminator loss.

### B. Related Works

Several related works have proposed various image pre-processing approaches for enhancing OCR text extraction. These methods can be categorized into two distinct approaches. The first approach involves traditional image editing techniques applied before OCR text extraction [3, 6]. The second approach leverages machine learning, utilizing its capabilities to perform image understanding through semantic segmentation [4, 5].

Hengaju *et al.* [3] proposed image pre-processing pipeline to enhance text extraction in scanned image document and mobile phone camera for Nepali language using image editing technique. The text extraction process utilizing Tesseract OCR. The pipeline starts with performing illumination aspect adjustment to perform brightness correction and contrast stretching. With mobile captured image usually suffer from skewness, orientation-aspect correction by performing skew, perspective, and warps correction. After that, the resolution aspect was also optimized, by resizing the image into 300 dpi [5]. The pipeline continues with image editing process by increasing the image contrast by performing monochrome conversion, noise removal, and image sharpening.

Christian *et al.* [6] proposed image pre-processing and post-processing to enhance text extraction in Indonesia KTP (Kartu Tanda Penduduk). The text extraction process utilizing Tesseract OCR. The pre-processing step consists of resizing the image with bicubic interpolation to smoothen and enlarge the image. After that had its background deleted. Lastly to convert the image coloration to greyscale. The process then continues with text extraction. The result of OCR, then being proceed employing look-up table on certain part of the KTP card, such as provinces, regions, religion, occupation, citizenship status, and marital status, employing N-gram and levenshtein distance, to replace any mis-extracted text based-on the lowest score.

Kumar *et al.* [5] proposed a pipeline in form of software that mainly focuses on image pre-processing technique before text extraction performed utilizing Tesseract OCR. The pipeline starts with retrieving input from scanner. With the OCR accuracy drops after a character had size below 20 pixels height, the image is required to be resized to 300 dpi. The pipeline continues with image editing process by

increasing the image contrast, perform monochrome conversion, noise removal, and de-skewing text. Finally, to perform document layout analysis to detect various basic elements in an image employing machine learning. However, the mentioned machine learning technique remains undisclosed.

Fleischhacker *et al.* [4] proposed image pre-processing step of semantic segmentation employing faster Region-based Convolutional Neural Network (R-CNN) with ResNet-50 backbone, before performing text extraction in 19<sup>th</sup> century historical documents utilizing Tesseract OCR. The proposed model can perform document layout analysis with different 8 classes, with a limit of 1000 objects per image.

Lat *et al.* [24] proposed image pre-processing using super-resolution technique to improve OCR accuracy by enhancing a low-resolution image, employing GANs. The proposed model demonstrates a 21% improvement in OCR accuracy when applied to image that reduced one-tenth of its original size, employing the modified Super-Resolution Generative Adversarial Network (SRGAN) architecture.

With previously several mentioned advancements in OCR enhancing techniques, we have found this interesting field of study keeps evolving. The initial stages of advancement focused on constructing a robust pipeline to enhance images through management of properties such as color, luminance, contrast, distortion, and blurs, which are important for text readability [3, 5, 6]. As machine learning, particularly deep learning began to revolutionize the field of computer vision, the research towards OCR enhances through semantic segmentation and image resampling technique to partition text extraction area for minimal text extraction errors [4, 5, 24]. This advancement is game changing due to artificial neural networks capability to perform image understanding.

However, the unanswered question in this field remains. While the previously mentioned works have greatly improved OCR text extraction through unique experimental approaches, the image document they work with remain in human readable level. These documents frequently suffer from common issues such as distorted or blurry text. Our works, however, goes in a different direction. We utilize the original image without any intermediate preparation operations instead of performing image resizing to achieve a specific still-readable dpi, as align with Kumar *et al.* [5] that stated as the OCR accuracy will dropped after a character had size below 20 pixels height. This causes substantial information loss to the image, which aligns with our goal of restoring the heavily degraded images to achieve levenshtein distance reduction through character error rates.

In the introductory section, we highlight our utilization of a deep learning-based approach. Specifically, we employ GANs architecture to perform image reconstruction through pixel-to-pixel up-sampling. Further details regarding this methodology will be presented in the Proposed Method chapter.

### III. PROPOSED METHODS

Our works explore image pre-processing techniques to enhance OCR text extraction leveraging GANs. In common OCR applications, the process of text extraction is typically executed in two-step approaches as shown in Fig. 4. The

initial step, which is *image input process*, involving the image input retrieval within the application as an interface for the user to upload the file to the back-end system. The second step, the *text extraction process*, is to activate the OCR and perform a text extraction to the uploaded image. Subsequently, the application then shows the result of text extraction to the interface. When it comes into image pre-processing technique, which referred as *image enhancement*, is an additional step between the mentioned steps above is

applied. The *image enhancement* super-resolved the image employing image super-resolution technique. This intermediary step functions as an intercepting system, where the image pre-processing is performed within an agnostic system that retrieves the uploaded file from the *image input process*. This procedure ensures that the OCR system receives an enhanced image prior to the *text extraction process*.

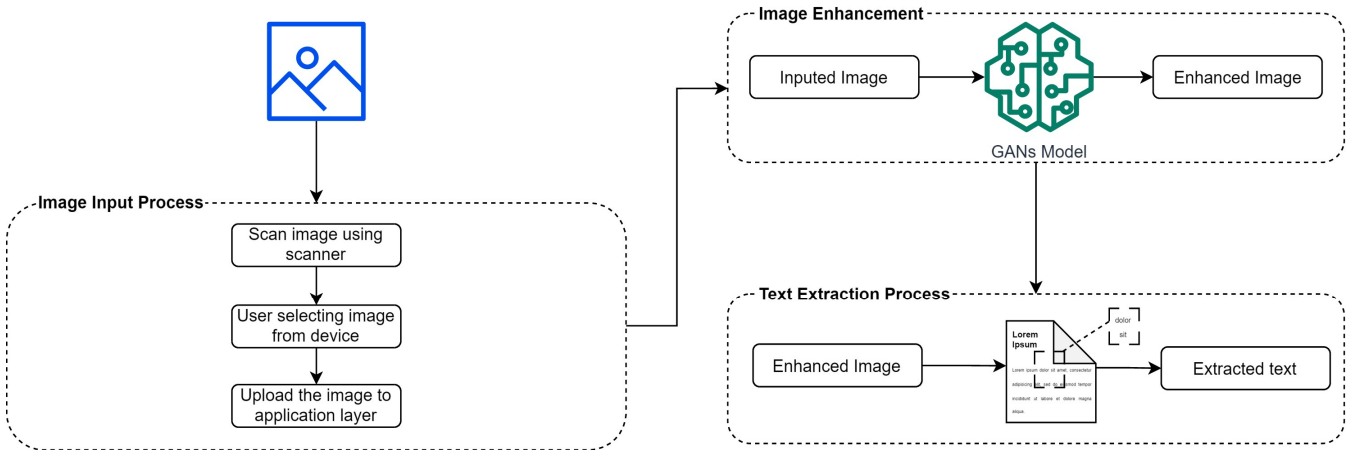


Fig. 4. Image enhancement process as intercept between image input and the OCR.

The *image enhancement* process involves a super-resolution task, employing the deep learning approach. Specifically, a GANs-based model is utilized. Due to the implementation of deep learning approach, it is important to train the model, enabling it to learn super-resolving image. This phase requires super-resolution dataset, model fine-tuning, and model testing process. This chapter covered the model building process through experimentation which involves selecting data for training the model, fine-tuning, and testing the model.

### A. Dataset Preparation

To train and evaluate the efficiency of the GANs model, the DocLayNet dataset, containing 80,863 PNG-format document images was employed [25]. This dataset consists of images with size of 1025×1025 pixel. The decision to employ this dataset, based on the data gathering technique described as by scanning the document. Which is aligned with the main required criterion to meet the premise of this research. Additionally, the dataset dominantly containing English written corpus. The sample of DocLayNet dataset is shown in Fig. 5.

From the initial dataset, a selection process involved identifying images that predominantly contained text was employed, to ensure the model learns relevant features for more accurate super resolution objective task. This partitioned the dataset with 1000 images for training, 200 for validation, and another 200 for testing.

Considering that text within down-sampled images will be very difficult for human visual perception and qualitative reading, we introduced an additional data segment beyond the previous training, validation, and testing sets. The data selection process for this segment is randomized, facilitating generability testing for our model. From now, this specific dataset segment is called qualitative testing dataset.



The objective of the award criteria is to evaluate the tenders with a view to choosing the most economically advantageous tender.

Tenders will be evaluated on the basis of the following:

1. Quality: **60 points**

The quality of the tender will be evaluated based on the above qualitative award criteria for the respective Lot.

2. Price: **40 points**

The price considered for evaluation will be the total price of the tender, covering all the requirements set out in the Tender Specifications, for the respective Lot.

The same will apply for the assessment of tenders under reopening of competition in LOT 1.

#### 3.5 Financial Award Criterion

Tenderers shall complete the Financial Offer Form in **Annex 12 "Price Catalogue"**.

The combination of prices/costs in the tenderer's financial offer shall include, account for and cover all costs that the tenderer may charge to ECHA in return for delivery of services under this FWC. It is the responsibility of the tenderer to ensure that all costs are incorporated into the Price Catalogue section.

Tenderers shall submit their prices as follows:

1. Price catalogue for profiles (both LOTS)
2. Price catalogue for basic RUN activities (only LOT 2)
3. Price catalogue for application management (only LOT 2)

The total price resulting from the sum of the total price of each price item above (as applicable per LOT) will be used for financial assessment purpose only.

Please note that all values and descriptions in the Price Catalogue that indicate the quantity or distribution of services that ECHA currently expects to request are approximate and tentative. ECHA does not undertake any commitment to place orders that reflect these values or descriptions.

The price offer from the winning tenderer will be annexed to the FWC and forms basis for the prices to be used in the context of specific contracts.

#### Detection of abnormally low tenders

Tenderers must be aware of Article 23 of Annex I to the Financial Regulation on abnormally low tenders. In order to make a consistency check of each tenderer's financial offer towards the level of service required, tenderers may be requested to provide a price structure document explaining in detail their pricing methodology. For further details, see Section 4.2 below.

Fig. 5. Sample DocLayNet dataset.

### B. Dataset Pre-Processing

To accommodate a model that objectively learns to do super resolution tasks, a data pre-processing process to generate high resolution and low-resolution pair of images are prepared from the data partition as shown in Fig. 6. This process starts with performing data standardization, by random cropping the image with size of 224×224 pixel. The standardization process required due to the model input training is required to be standardized in certain size. The random cropping process introduces the number of possibilities of obtaining different image samples from a single image. Which provide data augmentation, by

oversampling the data. This could be leveraged for achieving model generalization. While the high-resolution image refers to the result of the random cropping itself, the low-resolution image is created by performing down-sampling with 1:4 ratio scale from the random cropped image. Making a new image with size of  $56 \times 56$  pixel. Due to the nature of color channel value ranged from 0 to 255, a normalization process that reduces the standard deviation of the color channel value range into  $-1$  to  $1$  was conducted. This accommodates the required generator model output that utilizes hyperbolic activation function. Lastly, with our model using channel first enable in the model, we flip the image array format from Height-Width-Channel (HWC) into Channel-Height-Width (CHW).

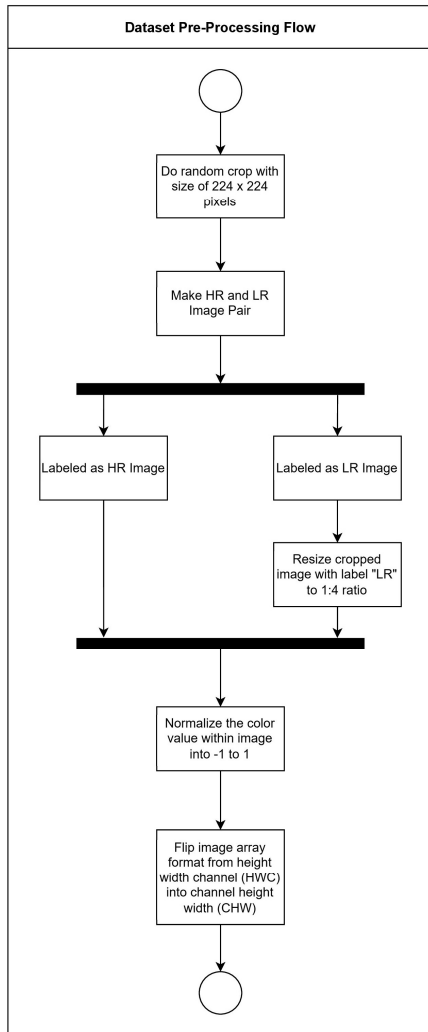


Fig. 6. Dataset pre-processing flow.

To ensure a comprehensive evaluation of the model performance, qualitative testing is also included. This testing phase aims to assess the quality of model from human perspectives both in text extraction and perceptual assessments. To achieve this, a separated data segment, distinct from previously training, validation, and testing sets was utilized for the qualitative testing phase.

The distinction of pre-processing process for qualitative testing lies in the generation of the low-resolution image method. The low-resolution image was generated using a

down-up-sampling method, which involves downsizing the image into a certain scale (which in our case, 25%), and force stretching the down-sampled image back to its original dimensions as shown in Fig. 7. This pre-processing method applied to augment document deterioration, where substantial information loss occurred on this document.

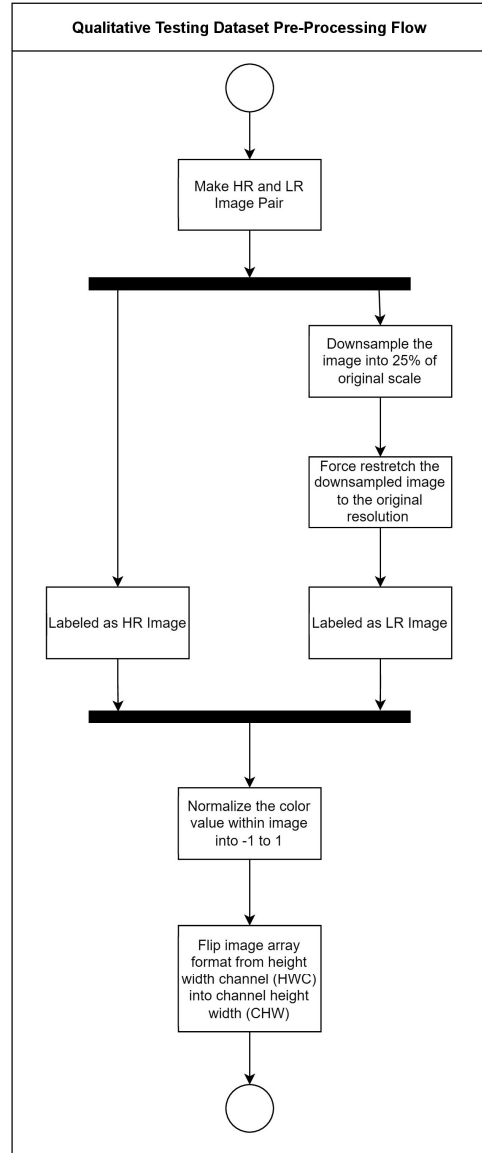


Fig. 7. Qualitative testing dataset pre-processing flow. The key difference lies in the force re-stretching the down-sampled image to the original resolution.

### C. Model Architecture

With a deep learning approach, the model is expected to perform super resolution tasks on low-resolution image. This is achieved through training a model to reconstruct or regenerate a low-resolution image, making it as close to or even better than its high-resolution counterpart. The underlying idea of building the model is to emulate human behavior by reimagining or redrawing a low-resolution image, resulting in an improved version that contains more readable text within the image. This process involves analyzing the entire low-resolution image to extract relevant information. Building a deep classification model, capable of determining



achieved by fooling the discriminator model by generating a convincing patch. With the required model specification above, we need to construct a discriminator model in the form of deep image classification network. The model outputs serve as an indicator of whether an image is genuine or fake [9]. To quantify this matter, the output layer of this model bounded between value of 0 to 1, where the higher bound corresponds to be classified of “fake”.

We explored Super-Resolution Generative Adversarial Networks (SRGAN) by Ledig *et al.* [13]. Which caught our attention due to its promising architecture and aligned well with our works. Consequently, we adopted their foundational model, and modified it to be aligned with our specific needs.

While working with a scanned corpus, getting craves of text is challenging considering the process of depicting images of characters. To tackle this problem, we increase the capacity of the convolutional layers within the generator model, by doubling the number of neurons, except for the output layer. However, we retained the residual network architecture flow, which has proven effective in self-capturing features. While adapting the discriminator model, we faced resource constraints due to the increased neurons count in our modified generator model. Consequently, we rebuild the entire discriminator model. The key changes here are replacing the BatchNorm2d layer into simpler BatchNorm layers. This significantly reduces GPU memory requirements. Also with these changes, we also altering the neuron size on each convolutional layer alongside its kernel sizes, guided by thorough feature extraction analysis. And finally, we applied Sigmoid activation function for the discriminator model. To furthermore save GPU memory usage, we adjusted the input layer dimensions. Subsequently, we reduce the original input for generator model, from  $96 \times 96$  to  $56 \times 56$  and the discriminator model, from  $384 \times 384$  to  $224 \times 224$ . The final model architecture is as shown in Fig. 8.

The training process of GAN model splits into two phases. The *initialization training*, and *adversarial training*. The purpose of *initialization training* is to pre-weight the generator model on performing super resolution tasks. The expected outcome of the first training phase is to make the model learn to fix the image by performing pixel-to-pixel reconstruction and remapping the image into  $4 \times$  scale after up-sampling. In endeavor producing such model, the *initialization training* phase uses similar technique as training regular super resolution convolutional neural network (SRCNN) [12]. To train the model, this phase uses using mean squared error (MSE) as loss function, by count the color aspect difference between the enhanced image and the ground truth. The image-related task contains uncertain and sparse gradients. Additionally, super-resolution task which initially training the model to work with low-resolution patch image, contains values that may cause the problem of local optima or local minima such as near-zero values, and can be troublesome mathematically. To facilitate and ensuring its generalizability, we conducted 300 training iterations (epoch), using Adaptive Estimation Moment (Adam) as optimizer with initial learning rate of  $0.02 (1 \times 10^{-2})$ .

The second phase is *adversarial training*. As the core of building GANs, this phase is taken to build model’s generative nature [13]. The pre-weighted generator model earlier was taken and proceed to perform adversarial training

to build model’s generative nature on performing image recreation after given low-resolution patch. As an adversarial training goal is to achieve *nash equilibrium*, where the discriminator model must not be able to distinguish the generated image, which is calculated using binary cross entropy. This judgment is called as adversarial loss. In addition of the adversarial loss, we also employed the pretrained Visual Geometry Group 19 (VGG19) model feature extractions for performing perceptual differences quantification solely named as content loss by Ledig *et al.* [13]. The result of both adversarial loss and content loss then being summed, producing perceptual loss [13]. This training phase also employed Adam optimizer with learning rate of  $0.02 (1 \times 10^{-2})$ . The *adversarial training* phase was conducted 3000 training epochs by performing training in batches of 500 epochs each. To prevent both local optima and minima phenomenon, we applied step decay after 350 epochs in every batch.

#### D. Model Training Evaluation Metrics

To accommodate the current configuration, which requires both training and validation set, we modified both training phases. For the initialization phase, we added validation phase and retain the usage of Mean Squared Error (MSE) for its metric. However, in the adversarial training, we introduce Structural Similarity Index Measurement (SSIM) and Peak Signal-to-Noise Ratio (PSNR) [16, 17]. These metrics serve to quantitatively score the model capabilities on performing super resolution tasks. The SSIM measures the model’s effectiveness on recovering the main structure of the document [18]. The PSNR evaluates the model’s ability to reconstruct the low-resolution image [19]. However, both SSIM and PSNR are positive correlatives metrics where lower score evaluation stands sign a low quality. This is contradictory with back propagation process that is aim for minimization problem. The possible choice to adapt is Mean Squared Error (MSE) by performing pixel-to-pixel comparison, however it struggles when quantifying the qualitative aspect of the image such as perceptual aspect. Therefore, we adapt the perceptual loss by Ledig *et al.* [13, 16] to be accounted as loss metric during adversarial training. In our works, due to the discriminator model applying sigmoid activation function in the output layer, we remove the required multiplication of 0.001 for the discriminator loss. All metrics are required to perform a side-by-side comparison of generated image with its high-resolution image counterparts.

#### E. Evaluation Techniques

After training the model to perform super resolution task and assessing its performance based-on perceptual aspect, we then evaluated the model based-on how effective the OCR system could read the enhanced image. The evaluation techniques involve the trained generator model, employing the remaining testing set. The general purpose of testing dataset is to evaluate the aspect of model generalization; therefore, we evaluate the model employing SSIM, PSNR, and levenshtein distance. Which splits the testing phase into two, the image similarity metrics and the levenshtein distance.

As shown in Fig. 9, the image similarity evaluation employed both SSIM and PSNR to quantify the similarity between the enhanced image and its high-resolution

counterpart. The SSIM metric, as its name “structural”, accounts the structural differences aspect by accounting image color covariance, which resemble luminance and contrast of both images [18]. The higher the SSIM score (max 1,0), indicates greater structural similarity. The PSNR is derived from the MSE, which evaluates the image overall similarity [19]. Unlike SSIM which focus on structural aspect, PSNR measures differences in image colors, and represent it into decibel (dB). The higher the PSNR score indicate the more similar it is.

The levenshtein distance, also known as edit distance, is a measurement technique that quantifies editing operations required to transform one string into another such that it matches the ground truth. The editing operation includes insertion, substitution, and deletion as formulated in Eq. (7) [11]. To elevate the utilization of this metric, we introduced Character Error Rate (CER) where the value of levenshtein distance is divided by the amount characters in the ground truth as formulated in Eq. (8).

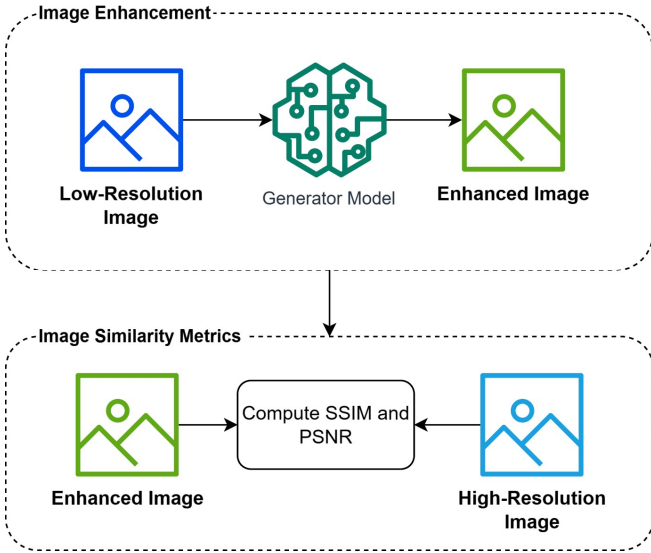


Fig. 9. Image Similarity Metric evaluation phase flow.

#### 1) Levenshtein distance formula

$$lev_{x,y}(i,j) = \begin{cases} \max(i,j), \min(i,j) = 0, \\ \min \begin{cases} lev_{x,y}(i-1,j) + 1 \\ lev_{x,y}(i,j-1) + 1 \\ lev_{x,y}(i-1,j-1) + 1_{(x_i \neq y_j)} \end{cases}, \text{ else.} \end{cases} \quad (7)$$

where:

$x$  stands for text output from fixed image.

$y$  stands for text output from ground truth.

$lev$  stands for edit operation required.

$i$  and  $j$  stands for current character position of string  $x$  and  $y$ , respectively.

#### 2) Character error rate formula

$$CER(x,y) = \frac{lev_{x,y}}{\text{total character in } y} \quad (8)$$

where:

$x$  stands for text output from fixed image.

$y$  stands for text output from ground truth.

$lev$  stands for levenshtein distance.

To accommodate all requirements above, a testing methodology employing OCR was conducted by building

process that emulates the proposed methods, which shown in Fig. 10. This is by deploying the trained generator model and employing it as an image pre-processing system. All low-resolution images are going through this mechanism, essentially generating fixed images, now referred to as ‘enhanced image’. Then the enhanced, low-resolution, and high-resolution images then are through the OCR for text extraction process. All of them had their text extracted, then both extracted enhanced and low-resolution texts have its CER score calculated employing the extracted high-resolution texts as ground truth.

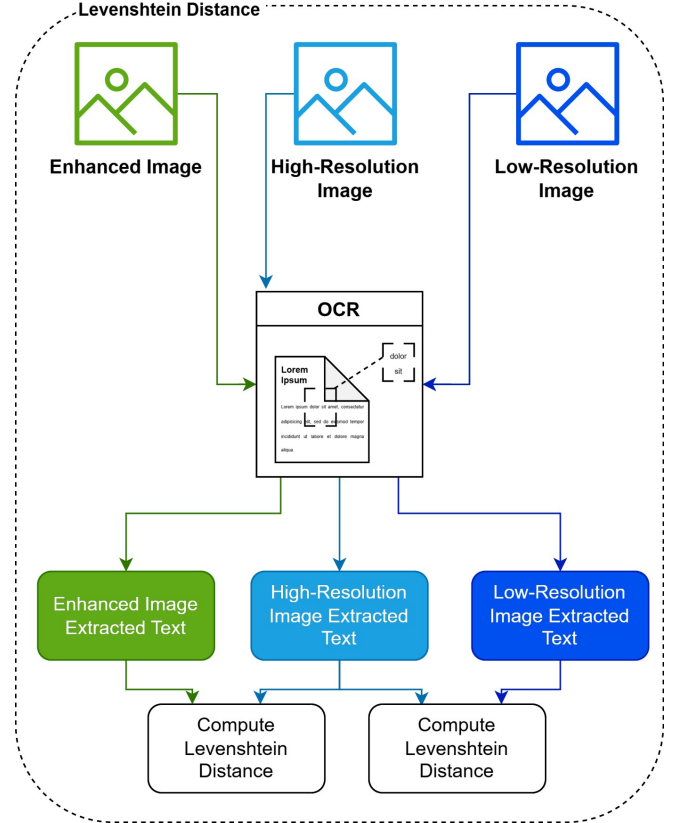


Fig. 10. Levenshtein Distance Metric evaluation phase flow.

## IV. RESULT AND DISCUSSION

The data pre-processing process generates low and high-resolution pair from the random cropped 224×224-pixel size image as shown in Fig. 11. While the high-resolution image was taken directly from the cropping result, the low-resolution image was made by performing 1:4 downscale from the cropping result. The low-resolution image then used as data for the model, and the high-resolution image as its label target. This process successfully augmenting the dataset by providing random factor with chance of 1:640000 as Eq. (9). With the high training data variance, we achieved generalization. The decision to perform a data selection, also evident in beneficial to hinder chance of random cropping process falls into picture area that contain full of whitespaces or no text.

Numbers of possible samples that can be generated from random cropping:

$$\text{total}_{\text{sample}} = (H_{\text{image}} - H_{\text{crop}} - 1) \times (W_{\text{image}} - W_{\text{crop}} - 1) \quad (9)$$

where:



Fig. 16. Low Resolution and Generated image text extraction result compared qualitatively.

We evaluated the performance of text extraction using Tesseract OCR to the enhanced image. With the initial average CER of 98% for the text extraction from the raw low-resolution image, which was made apparent that the image downscaling by 75% provides significant information loss. When performing super resolution tasks on low-resolution image, our approach involves up-sampling and recovering the object as effectively as possible. Specifically, our model employs 2 Subpixel layers with scaling factor of  $2\times$ , providing  $4\times$  image up-scaling.

For qualitative text extraction testing, we conducted by text extraction to the whole document. During this phase, we observed an interesting phenomenon of, despite the inherent difficulties presented by barely recovered images, in which perceptually harder to be read by human, Tesseract OCR performs better as shown in Fig. 16. The result showed that the part of text “Chapter 18 perlfaq2”, completely unreadable in low resolution. The output of the text extraction only shows the remaining garbled sequence of characters are part of the rest of the page due to information loss after down-up-sampling performed. This is due to the Tesseract OCR that is harder in identifying text when the text is blurry and noisy. In the resampled counterpart, the text is identified, shown by its text extraction result showing “hopter 42 pEvliaeg”. However, the generated image seems to successfully restructure the text, resample the characters layout despite wrong coloration of predominantly white. Another example was found when working with colored background. Due to font color choice, the decreased contrast due to image down-up-sampling, and the serif font-family choice, the low resolution image failed to identify the word “leading”, showing “lc.nlin: \_:” instead as shown in Fig. 17. However, the resampled version shows 1 mistake prior to the extracted text showed as “leacding” instead of “leading”.

Fig. 17. Qualitative text extraction testing for colored background page.

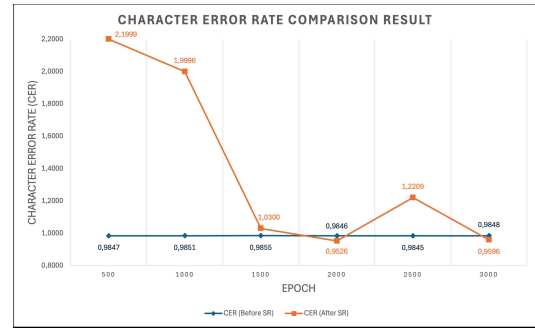


Fig. 18. CER evaluation comparison testing over 500-epoch training batches.

These results of the image enhancement process showed peak reduction of 3.20% CER score when compared with the low-resolution image. The quantitative comparison using CER measurement makes this evident, especially in epochs 2000 and 3000 as shown in Fig. 18. Interestingly, despite epoch 2000 achieving the best CER reduction of 3.20%, the PSNR score was worse than the epoch 3000. Additionally, epoch 1500 shows a worse CER result of 4.45% increase, while its PSNR score peaked at 20.7596 dB. This finding suggests that epochs 1500 and 2000 suffer in local optima problem which is bad for generalibility. Which is solved after conducting more iterative learning by training it to epoch 3000. Respectively, in big perspective aspect and in-detail perceptual aspect. The epoch 3000 however, despite only reducing the CER by 2.52%, shows better perceptual fixes both big perspective and in-detail perceptual aspects. This suggests that the model thoroughly learned and start showing its efficiency in generalization.

While there are no similar works specifically targeting OCR enhancement for documents that degraded 75%, we conducted a comprehensive comparison with Super-Resolution Generative Adversarial Network (SRGAN) [13, 24]. In perceptual quality, our model outperforms SRGAN. When performing quantitative analysis, SRGAN scored 0.9988 and 19.2862 in SSIM and PSNR respectively while our works achieved an impressive 0.9993 and 20.6035 for SSIM and PSNR respectively. This result highlights the accuracy with which our model ability to restore visual information, even in severely degraded documents. However, it is essential to recognize SRGAN strength. Both our works and SRGAN architecture excel in preserving overall structural integrity, ensuring the main document features are reconstructed. However, SRGAN lack of in-detail of individual characters fixing, producing hazy abstract text as shown in Fig. 19.

Fig. 19. The sample comparison between our works and SRGAN by PSNR score.

## V. CONCLUSION AND FUTURE WORKS

Our works successfully demonstrated the effectiveness of

using Generative Adversarial Network (GANs) to enhance OCR text extraction through performing super resolution task on document image. However, we believe that there is still room for improvement.

Training an artificial neural network based model like GANs, having the training process utilizing backpropagation which mainly performing minimalization the loss score. The lower training loss score, indicate the model has figure out its objective. In our work, perceptual loss still employed as loss function, which proved to be able to reconstruct the overall structure of the image, capturing layouts such as shapes, textures, and patterns. But some text part still showed as garbled text. Therefore, experimenting through employing text-based metric as loss score holds promise for advancing this field.

Our works used the actual original image as the ground truth of the document to train the GANs model. With GANs capability to perform style transfer, to further advance this field, training the GANs model to perform image resampling using pre-enhanced original image as the ground truth holds promise for advancing image resampling performance. Specifically, by making the text bolder, pre-convert the image to grayscale, and de-skewing the image before employing it for training.

OCR tasks vary. While our works focus on digital printed-scanned documents, challenge remains in specific another domain such as handwritten document. Each domain presents unique challenges related to text extraction. Thus, developing domain-specific GANs models or fine-tune existing ones to handle specific scenarios holds promise for advancing this field.

As future contribution, our works can play a pivotal role in the information extraction pipeline. One of the difficulties in dealing with image-based type documents is the low quality that can impair its readability and pose a challenge for text analysis and extraction. Low quality means that the document may be distorted, blurry, faded, or have other issues that can affect the text extraction process from recognizing the text. This oftenly happen due to the document aging, or deteriorating scanning devices. To improve document readability, our works can be employed for image pre-processing phase before text extraction begin.

To facilitate reproducibility and further exploration, the source code for our research implementation is publicly accessible on GitHub. Interested readers can find the repository at the following link below: <https://github.com/yosua-kristianto/gan-document-restoration>

#### CONFLICT OF INTEREST

The authors declare no conflict of interest.

#### AUTHOR CONTRIBUTION

Yosua Kristianto: Conceptualization, Formal analysis, Investigation, Resources, Data Curation, Writing–Original Draft, Visualization. Benfano Soewito: Conceptualization, Methodology, Validation, Writing–Review & Editing, Supervision. All authors had approved the final version.

#### DATA AVAILABILITY

To increase the transparency of this paper, the authors have provided data sources. The data is available at the Zenodo Repository: <https://doi.org/10.5281/zenodo.15717935>

#### REFERENCES

- [1] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018. <https://doi.org/10.1109/MSP.2017.2765202>
- [2] W. Bieniecki, S. Grabowski, and W. Rozenberg, "Image preprocessing for improving OCR accuracy," in *Proc. 2007 International Conference on Perspective Technologies and Methods in MEMS Design*, 2007, pp. 75–80. <https://doi.org/10.1109/MEMSTECH.2007.4283429>
- [3] U. Hengaju and B. K. Bal, "Improving the recognition accuracy of tesseract-OCR engine on Nepali text images via preprocessing," *Advancement in Image Processing and Pattern Recognition*, vol. 3, no. 3, pp. 1–13, 2020.
- [4] D. Fleischhacker, W. Goederle, and R. Kern, "Improving OCR quality in 19th century historical documents using a combined machine learning based approach," arXiv Preprint, arXiv:2401.07787, 2024.
- [5] S. Kumar, M. Sharma, K. Handa, and R. Jaiswal, "Improve OCR accuracy with advanced image preprocessing using machine learning with Python," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 7, pp. 1026–1030, 2020. <https://doi.org/10.35940/ijtee.G5745.059720>
- [6] I. Christian and G. P. Kusuma, "Improving OCR performance on low-quality image using pre-processing and post-processing methods," *International Journal of Engineering Trends and Technology*, vol. 71, no. 6, pp. 396–405, 2023. <https://doi.org/10.14445/22315381/IJETT-V71I6P239>
- [7] S. Karthikeyan, A. G. S. de Herrera, F. Doctor, and A. Mirza, "An OCR post-correction approach using deep learning for processing medical reports," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 5, pp. 2574–2581, 2022. <https://doi.org/10.1109/TCSVT.2021.3087641>
- [8] S. Drobac and K. Lindén, "Optical character recognition with neural networks and post-correction with finite state methods," *International Journal on Document Analysis and Recognition*, vol. 23, no. 4, pp. 279–295, 2020. <https://doi.org/10.1007/s10032-020-00359-9>
- [9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020. <https://doi.org/10.1145/3422622>
- [10] T. Hegghammer, "OCR with tesseract, Amazon textract, and Google document AI: A benchmarking experiment," *Journal of Computational Social Science*, vol. 5, no. 1, pp. 861–882, 2022. <https://doi.org/10.1007/s42001-021-00149-1>
- [11] D. Castells-Rufas, "GPU acceleration of Levenshtein distance computation between long strings," *Parallel Computing*, vol. 116, 103019, 2023. <https://doi.org/10.1016/j.parco.2023.103019>
- [12] Z. Wang, J. Chen, and S. C. H. Hoi, "Deep learning for image super-resolution: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3365–3387, 2020. <https://doi.org/10.1109/TPAMI.2020.2982166>
- [13] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 105–114. <https://doi.org/10.1109/CVPR.2017.19>
- [14] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2015. <https://doi.org/10.1109/TPAMI.2015.2439281>
- [15] C. Dong, C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proc. Computer Vision–ECCV 2016: 14th European Conference*, 2016, pp. 391–407.
- [16] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Image quality assessment: Unifying structure and texture similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 5, pp. 2567–2681, 2020. <https://doi.org/10.1109/TPAMI.2020.3045810>
- [17] Z. Wang and A. C. Bovik, "Mean squared error: Lot it or leave it? A new look at signal fidelity measures," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98–117, 2009. <https://doi.org/10.1109/MSP.2008.930649>

- [18] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004. <https://doi.org/10.1109/TIP.2003.819861>
- [19] F. A. Fardo, V. H. Conforto, F. C. de Oliveira, and P. S. Rodrigues, "A formal evaluation of PSNR as quality measurement parameter for image segmentation algorithms," arXiv Preprint, arXiv:1605.07116, 2016.
- [20] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "Dive into deep learning," arXiv Preprint, arXiv: 2106.11342, 2021.
- [21] M. Ismail, C. Shang, J. Yang, and Q. Shen, "Supporting ANFIS interpolation for image super resolution with fuzzy rough feature selection," *Applied Intelligence*, vol. 54, no. 7, pp. 5373–5388, 2024. <https://doi.org/10.1007/s10489-024-05445-7>
- [22] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," arXiv Preprint, arXiv: 1511.08458, 2015.
- [23] B. Franci and S. Grammatico, "Training generative adversarial networks via stochastic nash games," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 3, pp. 1319–1328, 2023. <https://doi.org/10.1109/TNNLS.2021.3105227>
- [24] A. Lat and C. V. Jawahar, "Enhancing OCR accuracy with super resolution," in *Proc. 2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, pp. 3162–3167. <https://doi.org/10.1109/ICPR.2018.8545609>
- [25] B. Pfizmann, C. Auer, M. Dolfi, A. S. Nassar, and P. Staar, "DocLayNet: A large human-annotated dataset for document-layout segmentation," in *Proc. the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 3743–3751. <https://doi.org/10.1145/3534678.3539043>
- [26] H. Gholamalinezhad and H. Khosravi, "Pooling methods in deep neural networks, a review," arXiv Preprint, arXiv:2009.07485, 2020.
- [27] J. Bjorck, C. Gomes, B. Selman, and K. Q. Weinberger, "Understanding batch normalization," *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. International Conference on Machine Learning*, 2015, pp. 448–456.
- [29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv Preprint, arXiv:1409.1556, 2014.
- [30] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, vol. 9, pp. 249–256.
- [31] L. Borawar and R. Kaur, "ResNet: Solving vanishing gradient in deep networks," in *Proc. International Conference on Recent Trends in Computing: ICRTC 2022, 2023*, pp. 235–247. [https://doi.org/10.1007/978-981-19-8825-7\\_21](https://doi.org/10.1007/978-981-19-8825-7_21)
- [32] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark," *Neurocomputing*, vol. 503, pp. 92–108, 2022. <https://doi.org/10.1016/j.neucom.2022.06.111>

Copyright © 2025 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).