Image Classification System for Online Gambling Advertisements Using Computer Vision

Muhammad Equilibrie Fajria*1 Rinaldi*2
*School of Electrical Engineering and Informatics, Institut Teknologi Bandung (ITB), Bandung, Indonesia
Email: 1equilibrie.01@gmail.com 2rinaldi@staff.stei.itb.ac.id

***Abstract*—Online gambling is a growing phenomenon in Indonesia that has led to various negative consequences. The proliferation of online gambling is largely driven by a massive amount of advertising across different media, including text, video, and digital images. One effective method to combat this issue is to block these online gambling advertisements. This paper presents the development of an image classification system specifically designed to identify and block online gambling ad images.**

The system is built on an integrated and modular framework that includes a text extraction module, a text classification module, an image classification module, a fusion module, and a graphical user interface (GUI). We employed pre-trained models such as PaddleOCR and EasyOCR for text extraction, while the IndoBERT Base Uncased model was used for text classification. For image classification, we utilized pre-trained EfficientNet-B0 and ResNet-50 models.

Our experimental results show that the combination of the EfficientNet-B0 model with the IndoBERT (BERT) model, using PaddleOCR for text extraction, achieved the highest accuracy of 96.40%. This demonstrates the effectiveness of a hybrid approach that leverages both visual and textual features for the task of online gambling ad detection.

***Index Terms*—Online gambling advertisement, image classification, Convolutional Neural Network (CNN), EfficientNet-B0, ResNet-50, text classification, BERT, Optical Character Recognition (OCR), PaddleOCR, EasyOCR**

I. Introduction

Online gambling conducted through internet platforms such as websites and mobile applications, has emerged as a significant global issue. This activity encompasses a wide array of games, including virtual slots, poker, sports betting, and mahjong, where participants place real money or digital currency bets in the hope of securing financial gains. However, players often incur substantial losses rather than profits. Furthermore, many online gambling games are designed with algorithms that can be manipulated by operators, making consistent winning and continuous profit generation virtually impossible for players. Despite frequent and significant financial losses, a considerable number of individuals remain addicted and continue to engage in online gambling.

In recent years, online gambling has surged in popularity worldwide, with a particularly pronounced rise in Indonesia. According to Budi Gunawan, the Coordinating Minister for Political and Security Affairs, an estimated 8.8 million Indonesian citizens were involved in online gambling as of November 2024. This has resulted in a massive economic drain, with data from the Financial Transaction Reports and Analysis Centre (PPATK) revealing that online gambling transactions reached approximately IDR 283 trillion during the second half of 2024. Beyond economic losses, this phenomenon has led to severe social consequences, including family conflicts, divorce, and an increase in criminal activities committed to fund gambling habits. Consequently, addressing online gambling in Indonesia is a critical and urgent matter given its widespread negative impacts on society.

The rapid proliferation of online gambling is fueled by advanced technology and increasingly accessible internet access. Previously confined to physical locations like casinos, gambling can now be accessed by anyone, anywhere, and at any time via personal electronic devices such as computers, tablets, and smartphones. This has made online gambling more visible, accessible, and appealing to a broad demographic.

Another key factor contributing to its prevalence is aggressive promotion. Online gambling operators heavily advertise their platforms across various social media platforms, websites, and mobile applications to attract new users. These promotions often feature enticing incentives, such as registration bonuses, deposit bonuses, and promises of significant cash prizes. The advertisements take various forms, including text-based messages (e.g., unsolicited messages on WhatsApp, or comments on YouTube and Instagram) and digital images displayed on websites.

To combat this issue, a robust detection system is required. Image-based online gambling advertisements can be effectively classified using computer vision technologies. At the time of this research, there is a notable gap in the literature, as no studies have directly addressed the classification of online gambling ad images using computer vision. However, existing research on Convolutional Neural Networks (CNNs) for image classification, Optical Character Recognition (OCR) for text extraction from images, and text classification provides a solid foundation. By integrating these established techniques, it is feasible to develop a system capable of accurately classifying online gambling ad images for the purpose of detection and blocking.

II. Related Works

A. EasyOCR

EasyOCR is a notable open-source OCR library developed by JaidedAI in 2020 [1]. Designed for end-to-end text detection and recognition, EasyOCR provides a straightforward interface that simplifies the entire OCR workflow for users. It is an end-to-end framework that handles various pre-processing, mid-processing, and post-processing steps to enhance the quality of the final output. The library also includes built-in default models, which are utilized if the user does not specify a different model. The default models are CRAFT for text detection and a Convolutional Recurrent Neural Network (CRNN) for text recognition.

The general process within EasyOCR involves two main stages: text detection and text recognition. First, the pre-processing stage adjusts the format of the input image to be compatible with the detection model. The detection model then identifies the presence and location of text within the image, which is output as bounding boxes. In the mid-processing stage, these bounding boxes are refined and merged based on predefined criteria, such as bounding box merging tolerance. Finally, these refined bounding boxes are passed to the recognition model along with the input image. This process generates the recognized text and its corresponding confidence score. The post-processing stage formats the final output, adding the bounding box coordinates to the recognized text to provide a complete result.

B. PaddleOCR

PaddleOCR, an Apache-licensed open-source toolkit developed by Baidu, is a comprehensive solution for end-to-end OCR and document parsing [2]. This toolkit is designed to streamline document processing, from text detection and recognition to a full understanding of document structure. PaddleOCR is built upon three core components: PP-OCRv5 for multilingual text recognition, PP-StructureV3 for hierarchical document structure parsing, and PP-ChatOCRv4 for key information extraction. Our work focuses on the PP-OCRv5 component, a robust OCR system capable of recognizing various text scenarios, including printed text, handwriting, and multiple languages.

The PP-OCRv5 pipeline consists of four main modules: an image preprocessing module, a text detection model, a text line orientation classification module, and a text recognition model. The image preprocessing module prepares the input image by enhancing its quality and correcting distortions or orientations, such as document rotation. For instance, if an image is inverted or skewed, a lightweight orientation module based on PP-LCNet and the UVDoc unwarping model can correct it before text detection. Next, the text detection model identifies areas containing text. PP-OCRv5 uses a new backbone, PP-HGNetV2, which significantly improves performance over its predecessors. The model's feature representation is further enhanced through a knowledge distillation process from a larger teacher model (GOT-OCR2.0). Data augmentation strategies, including hard case mining and random synthesis, are also applied to improve the model's generalization capabilities across diverse real-world conditions, even for challenging cases like blurred or randomly positioned text. After text detection, the text line orientation classification module checks the orientation of each detected text line. If a line is upside down or slanted, this module automatically corrects its orientation, ensuring the subsequent recognition stage receives standardized input. Finally, the text recognition model identifies the text within the detected areas. This model uses a dual-branch architecture based on PP-HGNetV2. The GTC-NRTR branch, trained with an attention mechanism, focuses on robust character sequence modeling. In contrast, the SVTR-HGNet branch prioritizes inference speed using a CTC loss function. In practice, only the lightweight SVTR-HGNet branch is used for predictions, providing an optimal balance of speed and high accuracy.

C. BERT

Natural Language Processing (NLP) has been revolutionized by advanced language models, among which the Bidirectional Encoder Representations from Transformers (BERT) model stands out. Developed by Devlin et al. (2019), BERT is a Transformer-based language representation model designed to deeply understand context by leveraging information from both left and right directions within every layer [3]. Unlike previous unidirectional language models, BERT employs a Masked Language Model (MLM) strategy to enable a bidirectional understanding of word relationships. Additionally, BERT is trained using a Next Sentence Prediction (NSP) task to learn the relationships between sentences, which is crucial for downstream applications like question answering and natural language inference.

The architecture of BERT is a multi-layer, bidirectional Transformer encoder that uses a self-attention mechanism to capture global word context. The model is available in two common sizes: BERT-BASE, which has 12 layers, a hidden size of 768, and 12 attention heads (totaling 110 million parameters), and BERT-LARGE, with 24 layers, a hidden size of 1024, and 16 attention heads (340 million parameters).

To handle various NLP tasks, BERT's input must represent a single sentence or a pair of sentences as a single token sequence. The input representation for each token is constructed by summing three types of embeddings: token embeddings, segment embeddings, and position embeddings. Token embeddings represent words or sub-words using a WordPiece vocabulary of 30,000 tokens. Segment embeddings indicate whether a token belongs to sentence A or sentence B, allowing the model to distinguish between sentence pairs. Finally, position embeddings provide positional information for each token, as the Transformer architecture inherently lacks sequential awareness.

The training process for BERT involves two distinct stages: pre-training and fine-tuning. During the pre-training stage, BERT is trained on a massive text corpus (BooksCorpus and English Wikipedia) using the MLM and NSP tasks. For MLM, approximately 15% of the tokens in an input sequence are randomly masked, and the model must predict their original vocabulary IDs based on the surrounding context. For NSP, the model receives a pair of sentences, A and B, and must predict whether B is the actual next sentence that follows A in the original corpus. After this pre-training, BERT can be fine-tuned for specific tasks by adding a simple output layer. This final layer can be used for a wide range of NLP tasks, including text classification, named entity recognition, or question answering, while the core BERT architecture remains the same.

D. EfficientNet-B0

EfficientNet-B0 is the baseline model of the EfficientNet series, an architecture designed to simultaneously optimize both accuracy and efficiency with a significantly smaller model size and computational footprint compared to traditional CNNs. The core innovation of the EfficientNet family is a novel scaling method called compound scaling [4]. This method systematically and uniformly balances the three key dimensions of a neural network: depth, width, and resolution.

Unlike conventional scaling methods that typically increase only one dimension, compound scaling uses a compound coefficient to scale all three dimensions simultaneously at a fixed ratio. This balanced approach is crucial because improving only one dimension can lead to diminishing accuracy returns. To maximize the effectiveness of this scaling method, the EfficientNet-B0 baseline was developed through neural architecture search, which optimizes for both accuracy and Floating Point Operations Per Second (FLOPS). The fundamental building block of the EfficientNet-B0 architecture is the Mobile Inverted Bottleneck (MBConv) block, originally introduced in MobileNetV2 [5]. MBConv leverages a combination of depthwise separable convolution to reduce computational load, inverted residuals for efficiency, and an added Squeeze-and-Excitation (SE) module to enhance feature representation by adaptively weighting each channel. By applying the compound scaling method to this EfficientNet-B0 base, larger models—from EfficientNet-B1 to B7—are systematically created.

The architecture of EfficientNet-B0 consists of several stages of MBConv blocks with varying expansion ratios and kernel sizes, followed by a 1x1 convolution layer, a global average pooling layer, and a fully connected layer with a softmax activation for classification. This approach allows EfficientNet-B0 to achieve competitive performance on large datasets like ImageNet while remaining resource-efficient, making it well-suited for real-time applications and computationally constrained devices. Additionally, EfficientNet demonstrates excellent transfer-learning capabilities, achieving high accuracy on various other datasets, such as CIFAR-100 and Flowers, with significantly fewer parameters.

E. ResNet-50

Training very deep neural networks is challenging due to performance degradation caused by the vanishing gradient problem. The Residual Network (ResNet) architecture was specifically designed to overcome this issue by introducing residual learning [6]. ResNet-50, a prominent variant with a depth of 50 layers, adopts this core concept by incorporating skip connections or shortcut connections. These connections create a direct path that allows the input of a layer block to be passed directly to its output, bypassing the non-linear layers in between.

In a residual block, the non-linear layers are tasked with learning the residual function, F(x) = H(x) – x, which represents the difference between the desired output function, H(x), and the input, x. Instead of learning the complex function H(x) directly, the block learns the relative change, F(x), which is then added back to the original input to produce the block's output, F(x) + x. This approach simplifies the learning process and stabilizes the gradient flow during training, preventing the signal from fading in very deep networks. As a result, ResNet-50 and similar residual networks can be trained to a much greater depth without the accuracy degradation that plagues traditional architectures.

The ResNet-50 architecture is built upon bottleneck blocks. Each bottleneck block consists of a sequence of three convolutional layers: a 1x1 convolution to reduce dimensionality, a 3x3 convolution for feature processing, and another 1x1 convolution to restore the original dimensionality. The shortcut connection is an identity mapping by default but uses a 1x1 projection convolution only when the input and output dimensions differ. With a computational cost of approximately 3.8 billion FLOPs, ResNet-50 is significantly more efficient than older models like VGG-16/19, while using a considerably smaller number of parameters.

The overall structure of ResNet-50 begins with a 7×7 convolution layer and a pooling layer. This is followed by four main stages: Conv2_x with 3 bottleneck blocks (56×56×256), Conv3_x with 4 blocks (28×28×512), Conv4_x with 6 blocks (14×14×1024), and Conv5_x with 3 blocks (7×7×2048). Finally, a global average pooling layer and a fully connected layer are used for classification. This design enables ResNet-50 to perform efficient deep feature extraction with enhanced training stability.

III. Methodology

The developed solution consists of two main aspects: the dataset and the image classification system for online gambling advertisements.

A. Data Preparation

Our solution is built upon a custom image dataset composed of two classes: online gambling advertisements and non-online gambling advertisements. This dataset was created and curated to train and evaluate our classification models. Data preparation consists of 4 steps:

1) **Data Collection:** The images were collected automatically using a publicly available scraping library to gather a large volume of images from various search engines.
2) **Data Cleaning:** The scraped images were not all suitable for direct use. We performed a data cleaning process to ensure the quality and relevance of the dataset. Irrelevant or duplicate images were removed. For images that were relevant but contained irrelevant objects, those objects were cropped out or removed to create a cleaner dataset.
3) **Data Split:** The finalized dataset was then partitioned into a training set and a testing set for model training and evaluation. To ensure robust model performance and reduce bias, we employed k-fold cross-validation on the training data. The training set was divided into several segments, where one segment was used for validation

and the remaining segments were used for training. This method helps to obtain a more representative and objective evaluation of the model's performance. Hyperparameter tuning was performed using this same k-fold cross-validation approach to identify the best hyperparameters that yielded the highest average accuracy. Once the optimal hyperparameters were determined, the model was retrained on the entire training dataset before being evaluated on the separate test set, which serves as a benchmark for real-world performance.

4) **Data Preprocessing:** We applied specific preprocessing steps to the dataset, depending on the task. For text extraction, the only preprocessing step was to convert all images to the JPG format. For image classification, the preprocessing workflow was more extensive:
    a) All images were converted to JPG format and resized to a uniform 224 x 224 resolution.
    b) We applied data augmentation on the training data to make the model more robust and improve its generalization capability. Augmentation techniques included horizontal flipping, random rotation, and adjustments to brightness, contrast, and saturation.
    c) Finally, all images were converted and normalized to match the specific input format required by the pre-trained models.

B. System Architecture

The developed system solution consists of five main components: a Graphical User Interface (GUI), a text extraction module, a text classification module, an image classification module, and a fusion module. These modules were developed individually and then integrated into a single, cohesive system. The overall architecture is shown in Figure 1

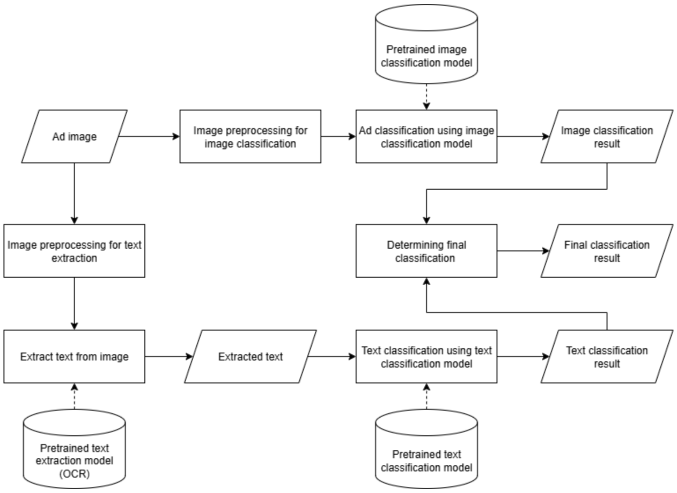
Pretrained image classification model
Ad image
Image preprocessing for image classification
Ad classification using image classification model
Image classification result
Image preprocessing for text extraction
Determining final classification
Final classification result
Extract text from image
Extracted text
Text classification using text classification model
Text classification result
Pretrained text extraction model (OCR)
Pretrained text classification model

Fig. 1. System architecture design.

The overall algorithmic workflow of the system is as follows:

1) An input image of an advertisement is received by the system.
2) This image is simultaneously fed into the text extraction module and the image classification module.
3) Both modules perform their respective preprocessing steps on the image.
4) The text extraction module extracts text from the image, while the image classification module classifies the image based on its visual features.
5) The extracted text from the text extraction module then becomes the input for the text classification module. This module classifies the ad image based on the content of the extracted text.
6) Finally, the classification results from both the text classification module and the image classification module are passed to the fusion module.
7) The fusion module combines these two classification results to determine the final class (online gambling ad or not) of the input image.

We will use PaddleOCR and EasyOCR for text extraction, BERT for text classification, and EfficientNet-B0 and ResNet-50 for image classification. To Determine the final classification, we will apply AND operator to both image classification result and text classification result.

C. Model Development and Evaluation

An online gambling advertisement image classification system cannot function without models. The text extraction module, text classification module, and image classification module all require ready-to-use models to operate effectively. However, not all of the necessary models for these modules are readily available. Only the text extraction models are ready for immediate use. The specific models needed for classifying online gambling ad images—for both text and image classification—do not exist yet. Therefore, we needed to develop new models for these two modules.

1) **Text Classification Model:** The text classification model was developed by fine-tuning a pre-trained IndoBERT model. IndoBERT is a variant of the BERT model that has been trained on an extensive Indonesian-language corpus, including Wikipedia, news articles, and web documents. The goal of this fine-tuning was to adapt the model specifically for the task of classifying text extracted from online gambling ad images.
For hyperparameter tuning, we utilized k-fold cross-validation on the training data. The training set was divided into several folds, where one fold was used for validation and the others were used for training. During each epoch of the training process, the model's accuracy, precision, recall, and f1-score were measured on the validation fold. The average scores across all folds were used to determine the set of hyperparameters with the best overall performance. The hyperparameter used for this process is shown in Table I
Once the optimal hyperparameters were found, the final model was fine-tuned using the entire training dataset. After the fine-tuning process was complete, the final model was evaluated on the separate test set to measure its performance, providing a final assessment of its accuracy, precision, recall, and f1-score.

TABLE I
BERT Hyperparameter Configuration

Hyperparameter	Configuration
Learning rate	2e-5, 3e-5, and 5e-5
Number of epoch	5
Batch size	16
Weight decay	1e-2
Warm up ratio	1e-1
Learning rate scheduler	Linear
Seed	42
Token max length	128
Optimizer	Adam

2) **Image Classification Model:** For the image classification task, we used two pre-trained models: EfficientNet-B0 and ResNet-50. Our approach involved fine-tuning both models to adapt them specifically for classifying online gambling ad images.
Just as with the text classification model, we applied k-fold cross-validation to determine the optimal hyperparameters for both EfficientNet-B0 and ResNet-50. The hyperparameter used for this process is shown in Table II. After identifying the best hyperparameters for each model, we proceeded to fine-tune them using the entire training dataset. During this fine-tuning process, all of the models' weights were trained on our custom image dataset. Once the fine-tuning was complete, the final models were evaluated on the held-out test set to measure their accuracy, precision, recall, and f1-score.

TABLE II
CNN Hyperparameter Configuration

Hyperparameter	Configuration
Learning rate	1e-3, 1e-4, and 1e-5
Number of epoch	20
Batch size	32
Weight decay	1e-4
Dropout rate	0.2 (EfficientNet-B0) and 0.5 (ResNet-50)
Criterion	BCEWithLogitsLoss
Optimizer	AdamW

IV. Results and Discussion

After developing and integrating each module and model, here are the evaluation result for text classification, image classification, and fusion module:

1) **Text Classification Module**: The evaluation of the text classification module began with validation during hyperparameter tuning for both the PaddleOCR BERT and EasyOCR BERT models. This process also indirectly validated the respective text extraction modules, as the text classification module's performance is directly dependent on their output.
Based on the validation results from the PaddleOCR BERT hyperparameter tuning, a learning rate of 5e-5 yielded the highest average accuracy at 96.90%, with a precision of 96.91%, recall of 96.90%, and an f1-score of 96.90%. The most significant change in average accuracy across all three learning rates occurred between epoch 1 and epoch 3. Ultimately, the 5e-5 learning rate was selected for training the final PaddleOCR BERT model. The trend of PaddleOCR BERT average accuracy changes during hyperparameter tuning can be seen in Figure 2

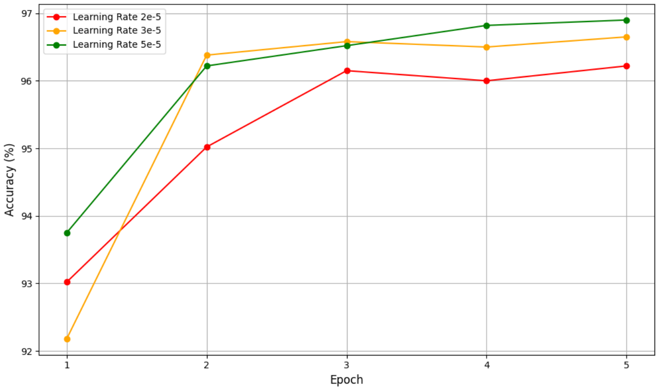

Fig. 2. Average accuracy change of PaddleOCR BERT model

Similarly, the validation results for the EasyOCR BERT hyperparameter tuning showed that the model with a learning rate of 5e-5 achieved the highest average accuracy at 96.25%, with a precision, recall, and f1-score of 96.25%. The most notable increase in average accuracy for all three learning rates was observed between epoch 1 and epoch 3. The learning rate of 2e-5 consistently recorded the lowest average accuracy across all epochs. Consequently, the 5e-5 learning rate was chosen for training the final EasyOCR BERT model. The trend of EasyOCR BERT average accuracy changes during hyperparameter tuning can be seen in Figure 3

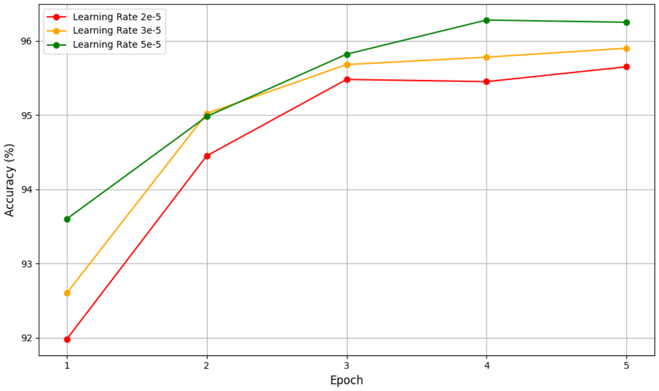

Fig. 3. Average accuracy change of EasyOCR BERT model

After selecting the optimal learning rate for both the PaddleOCR BERT and EasyOCR BERT models, final training was conducted using the entire training dataset. The performance of these final models was then tested on the unseen test dataset. The test result can be seen in Table III.
The test results revealed a difference in accuracy between the PaddleOCR BERT and EasyOCR BERT mod-

TABLE III
PADDLEOCR BERT & EASYOCR BERT TEST RESULT

Text Extraction	Accuracy	Precision	Recall	F1-Score
PaddleOCR	97.60%	97.60%	97.60%	97.60%
EasyOCR	96.81%	96.80%	96.80%	96.80%

els, indicating that the choice of text extraction model significantly impacts the text classification outcome. Upon analyzing the classification errors on the test set, we found that misclassifications were often caused by inaccuracies in text extraction. These inaccuracies typically occurred on images with text that was too small, used an unconventional font, had low color contrast, or was of low resolution. The errors from the text extraction could manifest as incorrect text location detection, misrecognized characters, unreadable characters, or incorrect character sequences.

Based on our evaluation using the collected dataset, PaddleOCR produced slightly superior text extraction results compared to EasyOCR, with the difference in accuracy being less than 1%. However, this minor difference is not sufficient to definitively conclude that PaddleOCR is absolutely superior to EasyOCR. A larger, more varied, and more representative dataset is required for a more accurate and conclusive comparison between the two text extraction models. Additionally, we observed a difference of less than 1% between the validation and test accuracies. This indicates that our model generalizes well and does not overfit to the training data, allowing it to perform consistently on new, unseen data.

2) **Image Classification Module:** The evaluation of the image classification module commenced with validation during hyperparameter tuning for both the EfficientNet-B0 and ResNet-50 models. Unlike the text classification module, the image classification module integrates feature extraction directly within the CNN model architecture. As a result, the testing of feature extraction and feature classification was conducted as a unified process.
Based on the validation results from the EfficientNet-B0 hyperparameter tuning, the model with a learning rate of 1e-4 achieved the highest average accuracy at 94.65%, with a precision of 94.44%, recall of 94.90%, and an f1-score of 94.67%. The model with a learning rate of 1e-5 showed the most stable change in average accuracy, followed by the 1e-4 and 1e-3 learning rates. This suggests that smaller learning rates tend to lead to more stable accuracy curves. Although the model with a 1e-5 learning rate had the lowest average accuracy from epoch 1 to 10, it demonstrated a consistent and upward trend in accuracy. Consequently, the 1e-4 learning rate was selected for training the final EfficientNet-B0 model. The trend of EfficientNet-B0 average accuracy changes during hyperparameter tuning can be seen in Figure 4
The validation results for the ResNet-50 hyperparameter tuning showed that the model with a learning rate of 1e-5 achieved the highest average accuracy at 94.75%, with

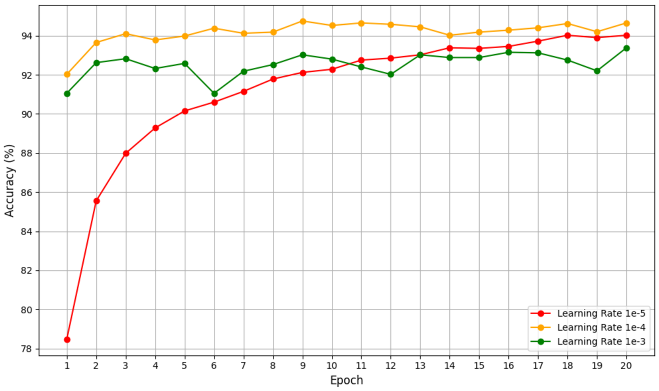

Fig. 4. Average accuracy change of EfficientNet-B0 model

a precision of 94.38%, recall of 94.20%, and an f1-score of 94.77%. Similar to EfficientNet-B0, the model with a 1e-5 learning rate exhibited the most stable change in average accuracy, while the 1e-3 learning rate had the most unstable performance and consistently lower accuracy. This indicates that a learning rate of 1e-3 is too high and unsuitable for training the ResNet-50 model. Ultimately, the 1e-5 learning rate was chosen for training the final ResNet-50 model. The trend of ResNet-50 average accuracy changes during hyperparameter tuning can be seen in Figure 5

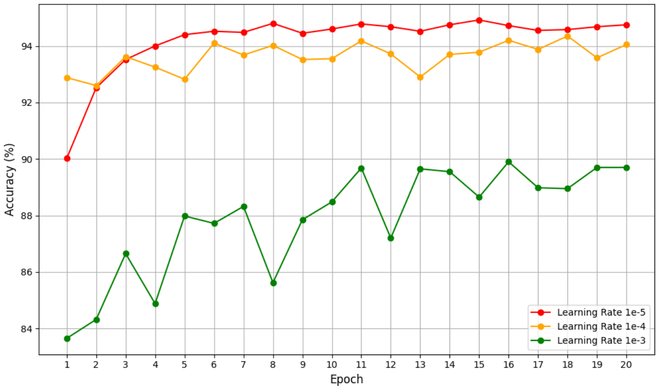

Fig. 5. Average accuracy change of ResNet-50 model

After selecting the optimal learning rates (1e-4 for EfficientNet-B0 and 1e-5 for ResNet-50), both models were trained on the full training dataset. Their performance was then evaluated using the test set. The test result can be seen in Table IV.

TABLE IV
EFFICIENTNET-B0 & RESNET-50 TEST RESULT

CNN Model	Accuracy	Precision	Recall	F1-Score
EfficientNet-B0	94.00%	92.80%	95.40%	94.10%
ResNet-50	94.10%	93.49%	94.80%	94.14%

Based on both validation and testing results, the ResNet-50 model demonstrated higher overall accuracy than the EfficientNet-B0 model. However, the difference was not significant, measuring less than 1%. Therefore, it cannot be stated with absolute certainty that ResNet-50 is superior to EfficientNet-B0, although it showed a slight edge on our collected dataset. A larger, more varied, and more representative dataset is needed to provide a more robust and conclusive assessment of the models' performance in real-world scenarios.

Furthermore, the difference between the validation and test accuracies for both models was also less than 1%. This indicates that both EfficientNet-B0 and ResNet-50 models generalize well and do not overfit to the training data. Through testing on the test set, we found that both CNN models had a higher rate of false negatives and false positives compared to the OCR-BERT models. This is likely because some non-online gambling ads share similar characteristics with online gambling ads, such as game top-up ads, and vice versa. Ads that possess characteristics of the opposing class are highly susceptible to be misclassified. This represents a key limitation of using CNN models alone for online gambling ad image classification, as online gambling ads can visually resemble non-online gambling ads.

3) **Fusion Module:** The fusion module was tested by combining the classification results from both the text classification module and the image classification module. The test result can be seen in Table V.

TABLE V
FUSION TEST RESULT

Fused Model	Accuracy	Precision	Recall	F1-Score
EfficientNet-B0 & PaddleOCR BERT	96.40%	99.79%	93.00%	96.27%
EfficientNet-B0 & EasyOCR BERT	96.10%	99.36%	92.80%	95.97%
ResNet-50 & PaddleOCR BERT	96.00%	99.57%	92.40%	95.85%
ResNet-50 & EasyOCR BERT	96.00%	99.57%	92.40%	95.85%

Based on our tests, the fusion model combining EfficientNet-B0 and PaddleOCR BERT achieved the highest accuracy compared to all other model combinations. This hybrid approach also proved effective in reducing the number of false positives. This is because the CNN model and the OCR-BERT model tend to make classification errors on different types of images, resulting in fewer online gambling ad images being incorrectly predicted as non-online gambling ads by both models simultaneously.

The fusion model also allows the CNN model to compensate for the weaknesses of the OCR-BERT model, which is heavily reliant on the presence of clear text. For instance, in cases where an online gambling ad image has unclear text, the CNN model can provide a more accurate classification than the OCR-BERT model.

Conversely, the OCR-BERT model can address the limitations of the CNN model, which relies on learned visual patterns from the training data. Some ads in circulation have unconventional visual patterns, making them more suitable for classification by the OCR-BERT model. The OCR-BERT model can handle these cases because the text found in online gambling ads and non-online gambling ads is often unique and easier to distinguish than other features, such as colors, object shapes, and object patterns.

However, a side effect of this fusion approach is an increase in the recall value. Compared to the individual models, the recall of the combined model is higher. This could be a significant drawback if it results in many non-online gambling ads being incorrectly detected as online gambling ads.

V. CONCLUSION

This study successfully developed and implemented a hybrid system for classifying online gambling advertisement images by integrating a text-based approach with a computer vision-based approach. We leveraged pre-trained models, including PaddleOCR and EasyOCR for text extraction, IndoBERT for text classification, and EfficientNet-B0 and ResNet-50 for image classification. Our results demonstrate that combining these two distinct classification methods significantly enhances the system's performance, achieving the highest accuracy of 96.40% with the EfficientNet-B0 and PaddleOCR BERT combination. This fusion approach effectively mitigates the individual weaknesses of each method, reducing the rate of false positives by leveraging both textual and visual features. The performance metrics across both text and image classification modules, with a less than 1% difference between validation and test accuracies, indicate that the models generalize well and are not overfit to the training data. This work provides a robust and promising solution for the automated detection of online gambling ads, which is a critical step in addressing their negative societal impacts.

VI. FUTURE WORK

Several potential improvements and enhancements can be made to the implemented online gambling ad image classification system:

1) **Refining Text Extraction Results:** Implement additional post-processing techniques to clean and correct the extracted text, which would improve the accuracy of the text classification module.
2) **Expanding the Dataset:** Collect a larger and more representative dataset of real-world ads to better train and evaluate the models, ensuring their effectiveness in diverse and real-world scenarios.
3) **Exploring Alternative Hyperparameters:** Conduct a more comprehensive search for optimal hyperparameters for both the text and image classification models to potentially achieve higher performance.
4) **Implementing Advanced Fine-tuning:** Explore more sophisticated transfer-learning strategies for the CNN models, where the model is first pre-trained on a similar domain before being fine-tuned, potentially leading to better feature learning.

5) **Benchmarking Alternative Models:** Investigate other state-of-the-art models for text extraction, text classification, and image classification to find superior alternatives to the ones used in this study.
6) **Selective Text Extraction:** Develop a method to extract only the most crucial or relevant text from the images, which could streamline the process and improve text classification accuracy by eliminating noise.
7) **Improving Fusion Methods:** Explore more advanced fusion techniques, such as early or intermediate fusion, to more effectively combine the features and classification outputs of the different modules.

Acknowledgment

I would like to express my deepest gratitude to all those who have supported and contributed to the completion of this research.

First and foremost, I am immensely grateful to my supervisor, Rinaldi, for his invaluable guidance, continuous encouragement, and profound expertise throughout the entire process of this research. His insightful feedback and dedication were instrumental in shaping this work.

My sincere thanks also go to Rila Mandala and Dicky Prima Satya for their constructive criticism and valuable suggestions during the final defense, which significantly improved the quality of this paper.

I would like to thank ITB for providing the necessary resources and an excellent academic environment for me to conduct this research.

I am also grateful to my family, especially my parents, for their unwavering support, prayers, and endless love. Their encouragement was my greatest motivation.

Finally, I extend my heartfelt appreciation to my friends for their moral support and for making this journey a memorable one. This achievement would not have been possible without your support.

References

[1] JAIDED AI, "EasyOCR," https://www.jaided.ai/easyocr/, 2020.

[2] C. Cui, T. Sun, M. Lin, T. Gao, Y. Zhang, J. Liu, X. Wang, Y. Zhang, C. Zhou, H. Liu, Y. Zhang, W. Lv, K. Huang, Y. Zhang, J. Zhang, J. Zhang, Y. Liu, D. Yu, and Y. Ma, "PaddleOCR 3.0 Technical Report," *arXiv preprint arXiv:2507.05395*, 2025.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2019.

[4] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 97, May 2019, pp. 6105–6114. [Online]. Available: http://proceedings.mlr.press/v97/tan19a.html

[5] V.-T. Hoang and K.-H. Jo, "Practical Analysis on Architecture of EfficientNet," *Faculty of Engineering-Technology, Quang Binh University & School of Electrical Engineering, University of Ulsan*, 2022.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. [Online]. Available: https://doi.org/10.1109/CVPR.2016.90