Deep Learning to Extract Animal Images With the U-Net Model on the Use of Pet Images

Agus Perdana Windarto1,*, Indra Riyana Rahadjeng2, Muhammad Noor Hasan Siregar3, Putrama Alkhairi1

1Prodi Sistem Informasi, STIKOM Tunas Bangsa, Pematangsiantar, Indonesia
2Prodi Teknologi Komputer, Universitas Bina Sarana Informatika, Jakarta, Indonesia
3Prodi Ilmu Komputer, Universitas Graha Nusantara, Padang Sidempuan, Indonesia
Email: 1*agus.perdana@amiktunasbangsa.ac.id, 2riyana.irr@bsi.ac.id, 3noor.siregar@gmail.com,
3putrama@amiktunasbangsa.ac.id

Abstract—This article explores the innovative application of deep learning techniques, specifically the U-Net model, in the realm of computer vision, focusing on the extraction of animal images from diverse pet datasets. As the digital landscape becomes increasingly saturated with pet imagery, the need for precise and efficient image extraction methods becomes paramount. The study delves into the challenges posed by varying animal poses and backgrounds, presenting a comprehensive analysis of the U-Net model's adaptability in handling these complexities. Through rigorous experimentation, this research refines existing methodologies, enhancing the accuracy of animal image extraction. The findings not only contribute to advancing the field of computer vision but also hold significant implications for wildlife monitoring, veterinary diagnostics, and the broader domain of image processing.

Keywords: Deep Learning; U-Net Model; Animal Image Extraction; Computer Vision; Pet Images; Semantic Segmentation

1. INTRODUCTION

In the age of digital proliferation, our world is inundated with a multitude of pet images, reflecting the deep bond between humans and their animal companions. Pets, also known as beloved animals, are living creatures that are specifically kept by humans for the purpose of entertainment, security or emotional companionship [11]–[13]. Various types of pets can be loyal friends, from loyal and high-energy dogs, graceful and independent cats, to singing birds that add joy to the house. Reptiles, such as turtles or snakes, as well as small animals such as hamsters and rabbits, are also often adopted as pets.

This diversity creates a unique relationship between humans and animals, resulting in close bonds and providing happiness and a more colorful life for their owners [4]. From social media platforms to personal photo albums, these images capture the essence of joy, love, and companionship. However, amidst this visual abundance lies a challenging problem: how can we accurately extract animal images from this vast and diverse pool, especially focusing on our cherished pets? This question forms the crux of our research, exploring the potential of cutting-edge deep learning techniques, particularly the U-Net model, in addressing this challenge [5], [6].

The rise of smartphones, coupled with the ubiquity of social media platforms, has led to an explosion in the sharing of pet photos. As a result, the internet is now flooded with images featuring animals in various settings, poses, and backgrounds. While this digital trend brings joy to pet lovers worldwide, it also presents a significant computational challenge: how can machines discern, delineate, and accurately extract animal features from this diverse visual landscape [7], [8].

There are various effective algorithms to overcome the task of pattern recognition and feature extraction in images, one of which is Convolutional Neural Network (CNN), which has proven successful in various image processing applications [9]–[12]. CNNs use convolutional layers to identify patterns and features at varying levels of complexity, enabling better object recognition and information extraction from images [13]–[15]. Apart from that, algorithms such as Feature Pyramid Network (FPN) are also often used to overcome challenges in feature extraction on objects of different sizes.

FPN leverages hierarchical feature pyramids to achieve higher levels of detail on small objects and better detect objects at multiple scales. Apart from CNN and FPN, other methods such as Histogram of Oriented Gradients (HOG) have also proven effective in extracting image features for object detection. HOG focuses on the orientation distribution of pixel gradients in images, which is very useful in depicting the texture and shape of objects. The use of a combination of these algorithms can improve performance and accuracy in pattern recognition in various types of images, providing a more holistic solution to the task of visual information extraction.

A number of well-known algorithms and architectures that have made significant contributions to image processing include U-Net, AlexNet, MobileNet, and several others. AlexNet, as one of the pioneers of Convolutional Neural Network (CNN), played an important role in raising the popularity of deep learning for image recognition tasks. With its deep architecture and ability to identify complex patterns, AlexNet is frequently used in a variety of applications including image classification.

On the other hand, MobileNet was developed with a focus on computing efficiency, making it very suitable for low-power devices such as smartphones. The MobileNet architecture is designed to speed up the training and inference process on limited devices, making it a popular choice in mobile and Internet of Things (IoT)
applications. In addition, there are various other architectures such as VGG, Inception, and ResNet, each of which offers a unique approach to feature extraction and pattern recognition in images. The choice of algorithm or architecture depends on the specific requirements of the image processing task and the available computing resources.

Enter the U-Net model, a deep learning architecture that has gained widespread acclaim for its proficiency in semantic segmentation tasks. Its unique U-shaped architecture enables it to capture intricate patterns and details within images, making it particularly well-suited for the task at hand. By understanding the spatial relationships and contextual cues within pet images, the U-Net model holds the promise of significantly enhancing the accuracy of animal image extraction.

Previous research conducted by [16] showed that a physics-guided CNN (PGCNN) with a rectangular input shape and a rectangular convolution kernel worked better than a basic CNN with higher accuracy and smaller uncertainty. The feasibility of designing CNN parameters with physics-guided rules derived from bearing fault signal analysis has also been verified.

Meanwhile, the next research conducted by [17] showed that the ensemble model tended to provide the best results. Additionally, the best CNN ensemble, called EnsenbleDVX, consisting of all three CNNs, provided the best results by achieving an average accuracy of 97.7%, an average precision of 97.7%, an average recall of 97.8%, and the average F1 score was 97.7%. Our research delves into the complexities associated with animal image extraction. Animals, especially pets, exhibit a wide range of poses and expressions. Moreover, the backgrounds against which they are photographed can vary drastically, from cluttered living rooms to expansive outdoor landscapes. These variations pose substantial challenges to traditional image processing techniques [18], [19].

However, the U-Net model, with its ability to comprehend the context of these diverse scenarios, offers a potential solution. By understanding the nuances of different pet poses and backgrounds, our study aims to fine-tune the U-Net model to handle these complexities effectively [20], [20]. This is where the how enters the picture. How can we navigate this immense sea of pet images with precision and efficiency. This is where the U-Net model steps in as the protagonist of our story.

Developed for semantic segmentation tasks, the U-Net model is uniquely tailored to comprehend intricate patterns within images. Its architecture, resembling the letter ‘U,’ allows it to capture fine details, making it ideal for discerning the subtle features of animals amidst varying backgrounds [21], [22]. By delving into the technical intricacies of this model, we explore the ‘how’ of our research, aiming to harness its potential for extracting meaningful insights from pet images [23], [24].

Beyond the realm of pet imagery, the implications of this research extend far and wide. Accurate animal image extraction has practical applications in fields such as wildlife monitoring, where researchers often rely on image data to track and analyze animal behavior. In the veterinary domain, precise image extraction can aid in diagnostics, allowing for detailed analysis of animal health [25], [26]. Furthermore, in the pet industry, innovations in image processing can enhance services such as pet grooming and training. In summary, our research endeavors to bridge the gap between the overwhelming abundance of pet imagery and the need for accurate image extraction. By leveraging the capabilities of the U-Net model, we aim to unravel the complexities of animal poses and backgrounds, thus paving the way for a new era in pet image processing [27], [28]. Through this exploration, we not only contribute to the advancement of computer vision but also enhance our understanding of the profound connection between humans and their animal companions in the digital age.

2. RESEARCH METHODOLOGY

2.1 Related Work And Deep Learning

Previous studies relevant to this research include various approaches in applying Deep Learning to extract animal images, especially using the U-Net model. Several previous studies have proposed innovative techniques in pet image processing, but there are still challenges in accurate feature recognition and extraction. These studies provide a basis for deep understanding of the complexity of this task, but it is important to note that the use of pet images as a specific focus has not been fully explored. Therefore, this research fills the knowledge gap by detailing the application of the U-Net model in overcoming these obstacles, with the hope of providing new contributions in the development of more sophisticated and effective animal image extraction techniques.

Deep Learning is part of the Neural Network which has a more complex architecture and a greater number of layers used, so it is expected to be able to handle more complex problems with more data and be able to produce fairly good output accuracy. A neural network algorithm that uses metadata as input and processes that input using a set of non-linear transformation functions arranged in layers and depth[29].

In Deep Learning there is a hidden layer whose job is to train a series of unique features based on the output of the previous network. This algorithm becomes increasingly complex and abstract when the number of hidden layers increases. Deep Learning’s neural network is formed from a simple hierarchy with several high-level layers or many layers (multi layers). Below in Figure 1 is an illustration of the difference between a simple neural network which only uses one hidden layer and Deep Learning which uses many hidden layers[30].
Figure 1. Deep Learning

Figure 1 explains that the CNN architecture is very complex, this can produce more accurate predictions and classification.

2.1.1 Neural Networks: Mimicking the Brain’s Architecture

Central to Deep Learning are artificial neural networks, inspired by the architecture of the human brain. These networks consist of layers of interconnected nodes, or artificial neurons, each layer contributing to the understanding of different aspects of the input data. The layers are divided into an input layer, one or more hidden layers, and an output layer. As data passes through the network, connections between neurons are strengthened or weakened based on patterns and relationships, allowing the network to learn and generalize from the provided examples.

2.1.2 Deep Architectures: Capturing Complexity

What sets Deep Learning apart is its ability to handle intricate and high-dimensional data. Deep architectures, with multiple layers, enable the learning of intricate features and representations from raw data. This is particularly advantageous in tasks such as image recognition, natural language processing, and speech recognition, where the complexity of the input data requires a hierarchical understanding.

2.1.3 Feature Learning: Uncovering Hierarchical Representations

Deep Learning excels in feature learning, a process where the model autonomously discovers relevant features from the input data. In traditional machine learning, engineers often manually design features for the model. However, in Deep Learning, the model learns to automatically extract hierarchical representations, reducing the need for explicit feature engineering. This not only streamlines the modeling process but also enhances the model’s adaptability to diverse datasets. In essence, Deep Learning is a transformative approach that has redefined the possibilities of artificial intelligence. Its capacity to autonomously learn intricate patterns from data has propelled breakthroughs in various fields, shaping a future where machines can comprehend, interpret, and make decisions on complex information.

This research method is designed to provide a comprehensive understanding of the comparison of CNN activation functions in pet type grouping. Thus, it is hoped that this article will make a valuable contribution to research in the field of artificial intelligence and its applications in the food industry, as well as provide practical guidelines for the optimal use of activation functions in pet type grouping. Comparing the performance of animal image classifiers using Convolutional Neural Network (CNN) architecture [2], [31]. The experimental design allows to test and compare several CNN architectures commonly used in image processing. The research design can be seen in the following picture [32]. Here’s a basic flowchart outlining the improvements made to a Convolutional Neural Network (CNN) using the U-Net architecture.

2.2 Dataset acquisition

In this study, the data set used includes pet data sourced from the website kaggle.com. The data is pet data in the form of cats and dogs.
Below is an example of a research methodology tailored to your study:

1. Start: Define the starting point of your research system.
2. Problem Identification & Data Collection: Define the problem you're addressing and collect the necessary data for your CNN.
3. Data Preprocessing & Augmentation: Prepare your data for training, including preprocessing steps like normalization and augmentation to increase dataset size and diversity.
4. Design CNN Architecture: Create the initial Convolutional Neural Network architecture tailored to your problem.
5. Training CNN: Train the initial CNN with the prepared dataset.
6. Testing and Evaluation: Test the trained model using the testing dataset. Evaluate the model's performance using metrics like accuracy, precision, recall, and F1-score.
7. Evaluate CNN Performance: Assess the performance of the initial CNN using appropriate metrics.
8. Identify Limitations & Areas for Improvement: Analyze the limitations of the initial CNN and identify areas that can be enhanced.
10. Training U-Net Enhanced CNN: Train the enhanced CNN, now incorporating the U-Net model features.
12. Compare Performance Metrics with Previous CNN: Compare the performance metrics of the enhanced CNN with those of the initial CNN to measure improvement.
13. Analysis and Results: Analyze the results obtained and draw conclusions about the effectiveness of the U-Net enhancements.
14. Conclusion & End: Summarize the findings and conclude the process.

2.3 U-net

U-Net is a proposed FCN based architecture. It has a u-shaped symmetrical design that includes an encoder, decoder, and bottleneck path. The encoder extracts the relevant features and these extracted features are propagated to the decoder using skip connections. After that, the decoder reconstructs the image to the desired dimensions using this feature map. The bottleneck path is placed between the encoder path and the decoder path, and contains two 3x3 convolution layers. There are four convolution blocks in the encoder path, and each block includes two 3x3 convolution layers with Rectified Linear Unit (ReLU) activation, and max 2x2 pooling layers. The decoder path also has four blocks, with each block having one 2x2 transposed convolution layer, a pooling layer to connect the extracted feature maps, and two 3x3 convolution layers.

2.4 CNN Architecture Development

We implemented a CNN model architecture for pet-based animal type grouping with image augmentation. The CNN architecture we use consists of several convolution and pooling layers, followed by a fully connected layer. We adjust the number of layers and filter size based on the characteristics of the data we have. We use Deep Learning libraries or frameworks such as TensorFlow or PyTorch to implement CNN models.

The U-Net architecture consists of two main parts: encoder (sequential part for feature extraction) and decoder (part for segmentation image reconstruction). The following is a detailed explanation of these two parts.

**Figure 7. Encoding Subnetwork**

In figure 7, the Encoder is responsible for extracting features from the input image. The U-Net architecture uses iterative convolution blocks for feature extraction. Each convolution block typically consists of multiple 3x3
(or 5x5) convolution layers followed by batch normalization and ReLU (Rectified Linear Unit) activation. These multiple convolution blocks are then followed by a 2x2 max-pooling layer to reduce image dimensions and increase computational efficiency. The encoder can consist of multiple levels, and each level extracts features with increasingly lower spatial resolution but with increasingly higher semantic representation.

**Figure 8. Decoding Subnetwork**

In Figure 8, the Decoder is responsible for restructuring the features that have been extracted by the encoder into a segmentation image. To that end, each decoder stage expands the spatial resolution of features by using up-sampling or transposed convolution (also known as deconvolution) operations. This operation helps recover spatial information lost during the encoder stage. After that, the features that have been obtained through the up-sampling operation are combined with the features from the equivalent encoder stage through layer merging (concatenation) to increase the semantic context.

**Figure 9. Model Optimasi Unet Dropout 0.3**
Figure 9 shows the model used in this research. Uses the Unet architecture which introduces a symmetric architecture and uses convolution techniques based on convolutional neural networks (Convolutional Neural Network, CNN) to handle image segmentation tasks well. The name "U-Net" comes from its general shape which resembles the letter "U". Symmetrical Network: U-Net has a symmetrical architecture, where the encoder and decoder parts have a similar shape allowing good information exchange between the two parts. Skip Connections: U-Net uses skip connections, where features from the encoder are directly copied and combined with layers in the decoder. This helps retain important spatial information that can be lost during convolution and up-sampling operations.

3. RESULT AND DISCUSSION

Based on the training results and testing results using different activation functions to compare the performance results of the best classifier as a pet analysis model as a test data sample.

3.1 Results

In the Loss graph it can be seen that the activation function in LogSoftmax tends to be better in the direction of training and testing movement. In this test, the researcher used Google Colab as a tool to test it, and the author also used early stopping which is useful for avoiding too large losses in testing rice quality. As a final result, you can see a comparison of several architectures used.

![Accuracy Graph](image1)

a. Model Dropout 0.3

![Loss Graph](image2)

b. Model Dropout 0.1

Figure 10. Model Optimasi Unet Dropout 0.3

Figure 10 shows the training and validation graph for image extraction of pet images. To overcome overfitting, we can try higher dropout values or apply other regularization techniques such as L1/L2 penalties on model weights. We can also try to change the model architecture, use data augmentation methods, or adjust other hyperparameters to improve model performance on validation data.

<table>
<thead>
<tr>
<th>Optimasi</th>
<th>Dropout</th>
<th>Train_Accuracy</th>
<th>Val_Accuracy</th>
<th>Loss</th>
<th>Val_Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adam</td>
<td>0.3</td>
<td>0.8997</td>
<td>0.8627</td>
<td>0.2528</td>
<td>0.3714</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>0.9087</td>
<td>0.8827</td>
<td>0.2328</td>
<td>0.3414</td>
</tr>
</tbody>
</table>

In Table 1 based on the research results you provided, there were two experiments carried out with changes in dropout values in the model using the Adam optimizer. I will help you explain the results of the research:
Experiment 1 (Dropout = 0.3):
1. Optimizer: Adam
2. Dropout Rate: 0.3 (30% of neurons dropped out during training)
3. Train Accuracy: 0.8997 (Training accuracy on training data)
4. Validation Accuracy: 0.8627 (Accuracy on validation data)
5. Train Loss: 0.2528 (Loss function value on training data)
6. Validation Loss: 0.3714 (Loss function value on validation data)

In this experiment, the model was trained with a dropout rate of 0.3. The training results show an accuracy of 89.97% on the training data, but it drops to 86.27% on the validation data. The loss function value on the training data is 0.2528, while on the validation data it is 0.3714. This indicates that the model may be experiencing overfitting, as there is a significant difference between training and validation accuracy.

Experiment 2 (Dropout = 0.1):
1. Optimizer: Adam
2. Dropout Rate: 0.1 (10% of neurons dropped out during training)
3. Train Accuracy: 0.9087 (Training accuracy on training data)
4. Validation Accuracy: 0.8827 (Accuracy on validation data)
5. Train Loss: 0.2328 (Loss function value on training data)
6. Validation Loss: 0.3414 (Loss function value on validation data)

In this experiment, the dropout rate was reduced to 0.1. Despite the lower dropout rate, the results show an improvement in accuracy on validation data (88.27%) compared to previous experiments. However, the difference between training and validation accuracy remains (2.6% difference), indicating that there is still a slight overfitting.

3.2 Discussion

The research results present a comprehensive evaluation of various optimizers to improve the performance of Convolutional Neural Network (CNN) models on the U-net architecture in the context of pet animal classification. These findings provide valuable insights into the impact of different training strategies on the model’s predictive ability and resource efficiency.

Figure 11. Pet image segmentation results
In this experiment presented in Figure 11, the dropout rate was reduced to 0.1. Despite the lower dropout rate, the results show an improvement in accuracy on validation data (88.27%) compared to previous experiments. However, the difference between training and validation accuracy remains (2.6% difference), indicating that there is still a slight overfitting. In segmentation, the prediction is almost the same as the original.

4. CONCLUSION

Based on the research results, it is evident that optimizing the dropout rate in the Adam optimizer significantly impacts the performance of the neural network. With a dropout rate of 0.3, the model achieved a commendable training accuracy of 89.97% and a validation accuracy of 86.27%, demonstrating its ability to generalize well. However, fine-tuning the dropout rate to 0.1 led to further improvements, yielding a higher training accuracy of 90.87% and an enhanced validation accuracy of 88.27%. This indicates that a dropout rate of 0.1 strikes a better balance between preventing overfitting and retaining valuable information within the model. Consequently, this research underscores the critical role of dropout optimization in enhancing the neural network's performance, emphasizing the importance of careful parameter tuning in deep learning tasks.

REFERENCES


Agus Perdana Windarto, Copyright © 2024, MIB, Page 475
Submitted: 20/12/2023; Accepted: 27/01/2024; Published: 28/01/2024


