Using Image Segmentation to Identify Firearms in CCTV Cameras

Anir Suren, Daniyal Ahmad Ansari, Blake Misquitta, Mário Rosa, Saaria Zaheer, Tae Esparanza Cooper

Mentor: David Jose Florez Rodriguez

Stanford University SHTEM Internship

Abstract

Segmenting firearms from complex backgrounds in images is a crucial task with applications in law enforcement and security, especially in a context where around 150,000 armed crimes have been committed only in 2022 (Statista Research Department, 2024). This study assesses the efficacy of machine learning for firearm segmentation and the possibility of a synthetic dataset supporting this task. CNNs excel at accurately segmenting and analyzing complex images but may face challenges related to data dependency and generalization. This study utilizes a dataset augmented from the Kaggle firearm segmentation dataset and a synthetic dataset created by superimposing firearm images onto diverse backgrounds of scenes. The results demonstrate significant potential to utilize image segmentation to identify firearms, with CNNs showing consistent performance. Future research on this topic should focus on expanding training datasets, exploring different image segmentation techniques and architectures, and optimizing models for real-time applications in diverse scenarios.

Introduction

In computer vision, there are three main ways of identifying objects or classifying images: image classification, which classifies the image based on identified objects; object identification, which identifies and locates objects; and image segmentation, which classifies the pixels of the image. For this paper, we used image segmentation to classify the gun pictures on our dataset images. At the same time, this technique is extremely useful in various fields of the health sciences, specifically in the area of medical imaging, where image segmentation can be used to identify different types of tissue during the cancer diagnosis process (MathWorks, 2024).

Another possible application of such structure, being the main focus of this paper, is the capability of identifying firearms. Such ability is extremely useful, especially in countries where the use of guns to commit crimes is recurrent, such as the United States, where in 2022 alone, there were 647 mass shootings (BBC, 2023).

Thus, this article aims to develop artificial intelligence that uses image segmentation to identify firearms with an artificial dataset.

Methodology

To build and implement an intelligence system capable of identifying and targeting firearms, models were programmed using *Python*, mainly on Google Colab notebooks, which used external GPUs with greater RAM and Disc sizes, such as TPU, which has over 334 GB of RAM and 225 GB of Disc.

Also, three significant steps had to be taken: organizing datasets; creating and choosing the model; and testing the selected model.

Dataset:

We trained models on two datasets:

· A dataset (N=66) adapted from an existing Kaggle firearm segmentation dataset (Weapons - Gun Detection & Segmentation, 2023) that included 11 images with segmentation masks (each image was augmented with two rotations (0° and 90°) and three transformations (none, left-right flip, and top-bottom flip), resulting in a total of 66 images.).

· A synthetic dataset (N=13,284) created by the study team by superimposing multiple images of 41 firearms in various orientations and positions onto 43 background scenes sourced from Pixabay, Pexels, and Unsplash. Images taken from these websites are in the public domain. Despite the considerable dataset size, only 1200 images were used, mainly because of model crashing issues. It is also worth mentioning that the resolution for those images was 256x256.

Figure 1: Example of image and mask in the synthetic dataset

The Model's Architecture :

All the created and tested models described below were Convolutional Neural Network (CNN) models, neural networks that have, normally, convolutional and pooling layers in their composition (Amidi et al, 2024). Four models were initially considered to explore various CNN architectures: three using pre-existing and already trained models from Keras and one mimicking a popular segmentation architecture. These were transfer learning models based on Efficient Net B0 (Tan et al, 2019), Mobile Net V2 (Sandle et al, 2018), Dense Net 121 (Albewi, 2022), and one using a simplified version of the u-net architecture (Ronneberger,2015). In all the transfer learning and U-net models, different dimensions for the kernels and different numbers of filters in the convolution layers were tested, with the dimensions (k) used for the kernels being 3x3, 4x4, and 5x5 and the number of filters (f) being 2, 4 and 8. It is worth noting that the activation function used in 3 out of the 4 models tested was the Rectified Linear Unit (ReLU). The ReLU activation function follows from the following mathematical expression: r(z)= max{0.0, z} ($\forall\ z\ \in$ R*+). This means that, for any number greater than zero, ReLU will behave linearly, whereas when a value for "z" is negative, the function will only result in 0. Given this property, ReLU becomes an excellent activation function as it brings simplicity and linearity to the convolution layers (Brownlee, 2020).

Nevertheless, ReLU has the downside of, sometimes, not activating neurons, which could lead to plateauing on essential metrics for this paper, such as IoU and Loss (CS231n, 2024). Because of that, we trained in 2 transfer learning models, which are listed below, with two different activation functions: Leaky ReLU, a function known for preventing the "dying" process of neurons, defined mathematically as $lr(x) = \{ x,\ \ if\ x > 0\ \ or\ \alpha x,\ if\ x \leq 0$ ($\forall\ x\ \in$ R) (*Papers With Code - Leaky ReLU Explained*, n.d.). For "α" values, 0.01, 0.1, and 0.2 were used, the most commonly used when using Leaky ReLU as an activation function.

ELU is the third activation function used and is known for being a good function for both positive and negative values in a dataset. ELU is mathematically defined as $ELU(y) = \{y, if\ y > 0\ or\ \alpha(e^{y} - 1), if\ y \leq 0$ ($\forall\ y\ \in$ R) (Krishna, 2018). For the values of "α," it was used 1.

In addition, the models' performance was analyzed using a batch normalization layer, which changes the mean of the internal model features made from the images to keep them closer to 0 and the standard deviation closer to 1 (Keras Applications, 2024).

As for the models that used the transfer learning technique, the first used the EfficientNetB0 base model; the second used MobileNetV2; and the third used Dense Net 121, all available on the Keras Applications platform. All were chosen based on their high Top-5 accuracy on the ImageNet dataset, 93.3%, 90.1%, and 92.3%, respectively; their inference time

per step (CPU), 46.0 ms, 25.9ms, and 77.1ms respectively; and, finally, their number of parameters, 5.3 million, 3.5 million, and 8.1 Million, respectively (Keras Applications, 2024).

Models were compiled with a custom dice loss function.

Models were trained on three epochs of 1,208 images from the synthetic dataset and tested on the augmented Kaggle dataset.

U-net model:

In the U-net model, we had two versions of the architectures: one with batch normalization layers and one without.

Transfer Learning Models:

As mentioned, it made three models using already trained models (Efficient Net B0, Mobile Net V2, and Dense Net 121). As shown in the GitHub code in the references section, all tree models had versions with and without batch normalization layers. They were all trained with a batch size of 32 through 5 epochs.

-Efficient Net B0: Was only trained using the ReLU activation function.

-Mobile Net V2: The ReLU and Leaky ReLU were tested as activation functions in this model. The code below was tested using the Leaky ReLU to reduce the plateau effect on the IoU metrics. It

-Dense Net 121: The ELU function was the activation function to prevent the “dying” of neurons problem.

Results

Training IoU scores ranged from 0.4426 (U-net model with batchnorm layers) to 0.6674 (Efficient Net transfer model). Similarly, testing IoU scores ranged from 0.4883 (U-net model with batchnorm layers) to 0.5783 (Efficient Net transfer model). Table 2 provides each model’s prediction of the mask shown in Figure 2.

Table 1: Intersection over Union (IoU) scores of different models trained on three epochs of 1,208 images from the synthetic dataset and tested on the augmented Kaggle dataset

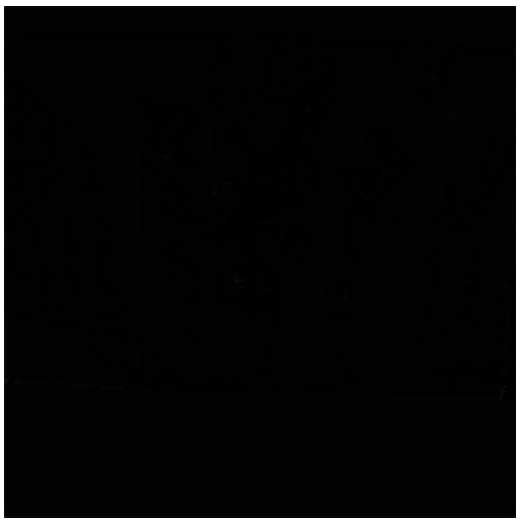
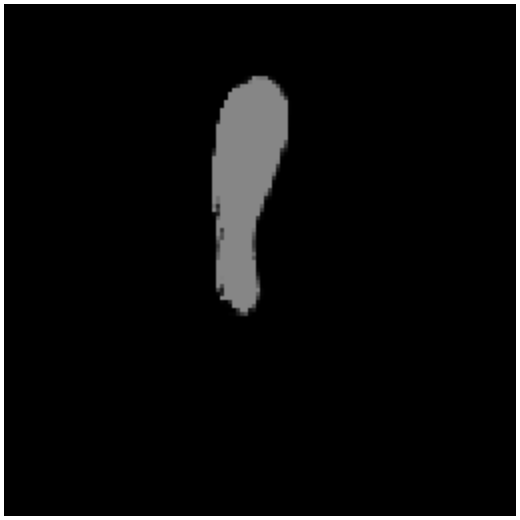
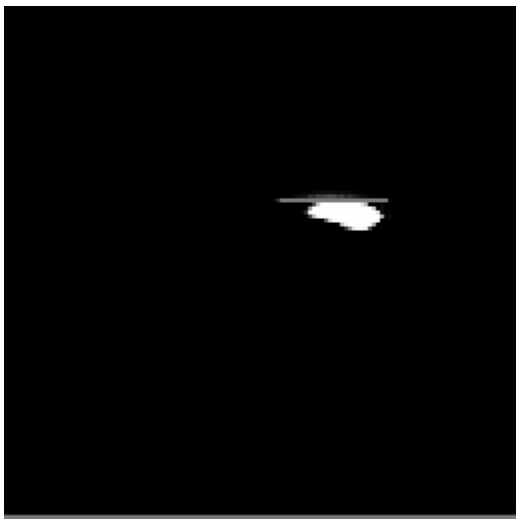
Model	Training IoU	Testing IoU
U-net model without batchnorm layers	0.4941	0.4954
U-net model with batchnorm layers	0.4426	0.4883

Efficient Net transfer model	**0.6674**	**0.5783**
Mobile Net transfer model	0.5017	0.5054

Figure 2: Image and mask from augmented Kaggle dataset used to demonstrate models' capabilities

Table 2: Predictions of different models trained on three epochs of 1,208 images from the synthetic dataset for an example image (Figure 2) from the augmented Kaggle dataset.

Model	Prediction
U-net model without batchnorm layers	

U-net model with batchnorm layers	
Efficient Net transfer model	
Mobile Net transfer model	

Discussion

Interpretation

The Efficient Net transfer model achieved the highest training IoU (0.6674) and the highest testing IoU (0.5783). This may be due to its pre-trained nature and large model size. However, masks generated by the Efficient Net transfer model are amorphous, limiting the model's predictive abilities. The U-net model without batchnorm layers lacks this issue, but it produces a scattered, disjointed prediction. The performance of both of these models may be improved through additional training.

The results from our study indicate significant potential for using image segmentation to identify firearms in complex images. Both GANs and CNNs demonstrated the ability to segment and identify firearms from various backgrounds with notable accuracy. GANs have robust unsupervised learning capabilities, produce high-quality segmented images, and effectively differentiate firearms from other objects. However, the instability and overfitting issues inherent in GANs were evident, especially when dealing with the highly diverse image sets.

CNNs, on the other hand, showed consistent performance in accurately segmenting firearms. The models using EfficientNetB0 and MobileNetV2, chosen for their high accuracy and efficient inference times, proved effective in this task. The application of transfer learning allowed these pre-trained models to adapt quickly to the new dataset, enhancing their segmentation capabilities. The U-net architecture also provided promising results, with its ability to capture intricate details in the images, further aiding in precise firearm identification.

Limitations

The quality of both datasets used is questionable. The Kaggle dataset, while containing real-world data, is quite limited in size; additionally, its masks are a color tint applied to the original image, and converting them to binary masks yielded imperfect results at times, with streaks showing up in the final mask. Conversely, our synthetic dataset is extensive but may not provide the quality needed for effective segmentation of real images.

Also, while the models performed well in a controlled environment, real-world applications like those present in CCTV footage may present more complex scenarios that these models were not tested against. Variations in lighting, image quality, and observance angle could affect the accuracy and reliability of this model’s firearm detection. This also raises the question of the computational resources required for training and deploying these models in real-time applications, which might pose practical challenges

Future Research

Our future research should focus on several key areas to address these limitations and enhance the effectiveness of image segmentation for firearm identification. First, increasing the diversity and size of the training dataset is important. Collaborating with law enforcement agencies or security agencies to access larger, more varied datasets that would improve model training and performance would be helpful. Experimentation with different image segmentation techniques and architectures, beyond GANs and CNNs, could also provide valuable insights. If possible, we could also explore hybrid models that combine the strengths of both architectures, which might yield better results. Additionally, investigating the potential of image segmentation to identify other objects, such as counterfeit currency or other concealed weapons, could expand the application scope of this technology. Furthermore, integrating these models into real-time surveillance systems requires optimization for speed and efficiency. Extensive field testing in various real-world scenarios is essential to validate the models' robustness and adaptability to different situations and circumstances.

Conclusion

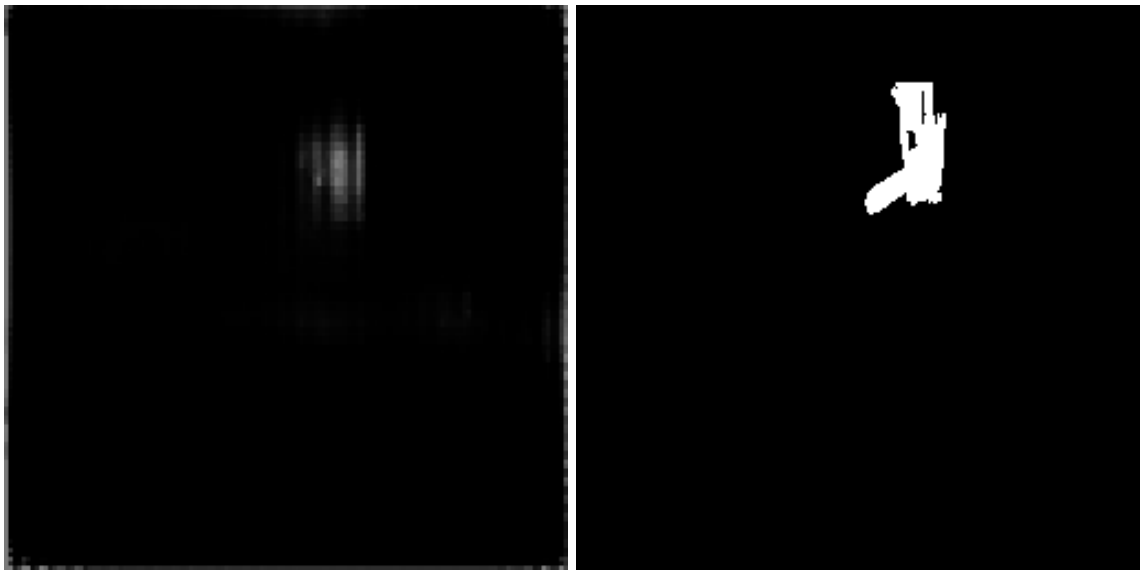


Figure 2: Efficient Net transfer model after two epochs of training on synthetic data set–predicted mask vs real mask

This research paper navigates the implications of image segmentations in identifying firearms. With rapidly increasing crime rates in many countries worldwide, this task is crucial for security.

AI Image segmentation could allow for greater clarity and efficiency in detecting and categorizing different image characteristics. This technique has been used in the biomedical field and has great potential for implications in law enforcement. It can help authorities recognize and locate firearms in images or CCTV camera footage, and ease the process of shortlisting suspects. This could significantly enhance the speed and accuracy of crime-solving, as well as emergency and precautionary actions by security agencies.

The paper focuses on using preexisting trained models, with 3 kinds of activation functions and 3 transfer learning models, on certain datasets to attempt to detect firearms through image segmentation. The models didn't perform as effectively as initially expected; however, they produced some good-quality images that distinguished firearms from the rest of the images, as seen in Table 2.

This study has used data from an existing Kaggle firearm segmentation dataset, and a synthetic dataset created by the team. Since the Kaggle dataset is restricted in size, while the synthetic dataset does not achieve the quality required, our data does not provide enough evidence to assure that the models would work reliably in real-world settings i.e. surveillance systems, where lighting and image quality can vary.

Nevertheless, the research can pave the way for future endeavors on this technology. By experimenting with different techniques, architectures, and hybrid models, collaborating with law enforcement for sufficiently diverse datasets, and testing these models in real-time, these could be improved for adaptation in endless defense, surveillance, and crime control possibilities.

References

- Gelana, F., & Yadav, A. (2018). Firearm Detection from Surveillance Cameras Using Image Processing and Machine Learning Techniques. In Smart Innovations in Communication and Computational Sciences (pp. 25–34). Springer Singapore. https://doi.org/10.1007/978-981-13-2414-7_3
- Brownlee, J. (2019, April 20). *A Gentle Introduction to the Rectified Linear Unit (ReLU) for Deep Learning Neural Networks*. Machine Learning Mastery. https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/
- Brownlee, J. (2020, August 20). *A gentle introduction to the rectified linear unit (ReLU)*. MachineLearningMastery.com. https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/
- *CS231n Convolutional Neural Networks for Visual Recognition*. (n.d.). Cs231n.github.io. https://cs231n.github.io/neural-networks-1/#actfun

- Team, K. (n.d.). *Keras documentation: Keras Applications*. Keras.io. https://keras.io/api/applications/
- Team, K. (n.d.). *Keras documentation: BatchNormalization layer*. https://keras.io/api/layers/normalization_layers/batch_normalization/
- *How many US mass shootings have there been in 2023?* (2023, December 7). BBC News. https://www.bbc.com/news/world-us-canada-41488081
- Amidi, S. & Amidi, A. (n.d.) *CS 230 - Convolutional Neural Networks Cheatsheet*. Stanford University. https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks
- Tan, M., & Le, Q. (2019, May 28). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. arXiv.org. https://arxiv.org/abs/1905.11946
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018, January 13). *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. arXiv.org. https://arxiv.org/abs/1801.04381
- Albelwi, S. A. (2022). Deep Architecture based on DenseNet-121 Model for Weather Image Recognition. *International Journal of Advanced Computer Science and Applications*, *13*(10). https://doi.org/10.14569/ijacsa.2022.0131065
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Lecture notes in computer science* (pp. 234–241). https://doi.org/10.1007/978-3-319-24574-4_28
- *Weapons - Gun Detection & Segmentation*. (2023, October 3). Kaggle. https://www.kaggle.com/datasets/trainingdatapro/people-with-guns-segmentation-and-detection
- *GitHub - ThePapayaInstitute/SyntheticImageSegmentationForFirearms: This year, our SHTEM lab worked on image segmentation. Focusing on the social issue of gun violence, we aimed to build a model that can detect and outline firearms in an image. Limited online datasets motivated us to ask how data augmentation and creating synthetic images could affect a model's performance on real data.* (n.d.). GitHub. https://github.com/ThePapayaInstitute/SyntheticImageSegmentationForFirearms
- *Papers with Code - Leaky ReLU Explained*. (n.d.). https://paperswithcode.com/method/leaky-relu
- Krishna. (2018, May 11). Introduction to Exponential Linear Unit - Krishna - Medium. *Medium*. https://medium.com/@krishnakalyan3/introduction-to-exponential-linear-unit-d3e2904b366c