Evaluation of Generic Deep Learning Building Blocks for Segmentation of 19th Century Documents

Evan Segal, Jesse Spencer-Smith, and Douglas C. Schmidt
Vanderbilt University, Nashville, Tennessee, USA
{evan.segual,jesse.spencer-smith,d.schmidt}@vanderbilt.edu

September 22, 2023

Abstract

Although the field of computer vision has grown significantly due to the advent of convolutional neural networks (CNNs), electronic analysis of historical documents has experienced scant research and development attention. Recently, however, computer vision has matured to the point where it can be applied to outperform existing, specialized tools for document analysis. This paper demonstrates empirically how state-of-the-art results can be produced by implementing, training, and evaluating generic computer vision models on historical document segmentation tasks. We show the generality of our approach to document analysis and explain how innovation in this domain can arise from combining generic building blocks for computer vision.

1 Introduction

Image segmentation is the process of partitioning an image by assigning a label or class to each of its pixels to represent the image meaningfully [7]. For example, an automated driving system may find it helpful to label objects in its environment, such as street signs and pedestrians, to assist with the driving process. Other examples of image segmentation may exist in content-based image retrieval systems [4], medical imaging [21], object detection [6], surveillance [19], generating data visualizations from hand-drawn sketches [29], and biometric security systems [11].

This paper explores methods and tools for image segmentation, specifically in the context of paper documents with handwritten records prior to the twentieth century. For the previously mentioned applications of image segmentation, there are many distinguishing features (such as color and brightness) between different objects. In historical documents, however, there is little/no color or contrast differences between parts of the document.

Instead, historical documents typically only exhibit logical differences that can be inferred from markings on the page, which are created inconsistently between records over time and are often degraded [16]. The segmentation of paper documents thus often necessitates different methods than the segmentation of other types of data. Moreover, the range of desired analysis on paper documents is extremely expansive, so it is important to consider the specific dataset we used in this paper and the problem that our analysis addresses.

This paper builds upon earlier work on dhSegment [3], which hypothesized that generic computer vision models could effectively perform document segmentation. We expand upon the dhSegement approach by (1) evaluating generic computer vision models other than the ResNet50-based [9] model used in dhSegment and (2) exploring what other models help advance this domain further using images found in the Slave Societies Digital Archive (SSDA)[1]. We hypothesize that the success of ResNet50 in classification tasks demonstrates its utility as a successful generic building block for constructing segmentation models compared to other common convolutional neural networks (CNNs).

The remainder of this paper is organized as follows:

Section 2 motivates and summarizes our technical approach; Section 3 summarizes work related to various types of significant CNNs used in our analysis; Section 4 describes adaptations to model architectures that we applied to enable CNNs to operate on segmentation tasks; Section 5 describes how we compared the ResNet-based dhSegment [3] to other CNN-based models by reviewing the experimental dataset, establishing a proof-of-concept and baseline for success, and elaborating on how experimental models are implemented and trained; Section 6 compares the results of training between similar models and between all of the best models, as well as analyzes trends that occurred during the training process; and Section 7 presents concluding remarks.

2 Motivation and Summary of Our Technical Approach

The motivation for this paper stems from the Slave Societies Digital Archive (SSDA) hosted at Vanderbilt University [1] that includes over 700,000 digital images drawn from ∼2,000 unique volumes dating from the sixteenth through twentieth centuries that document the lives of an estimated four to six million individuals. Slave societies are defined as civilizations where slave labor and/or trade was an essential part of their economies, politics, and lives as a whole. The SSDA preserves documents related to African people and their descendants in slave societies, mostly in the Iberian New World.

The majority of the documents in the SSDA are Catholic Church documents, which mandated the baptism of African slaves and their descendants. With baptisms comes eligibility for marriage and burial with the Catholic Church. Since the Catholic Church is a centralized, hierarchical organization, there is significant consistency between documents created in different parts of the world and at different periods in time.

Although the quality and the layout of documents may be different between record keepers, a base set of facts (such as) names, locations, dates, and the names of family members [1]) remains consistent throughout the documents. These common facts between documents create a structure that lends itself to algorithmic analysis rather than needing to analyze each of 700,000 images manually. It is not yet feasible, however, to simply extract the characters from the page using optical character recognition (OCR) technology [16] and then analyze the text. Instead, this information must be derived from other features to analyze the SSDA archive meaningfully.

The ultimate goal of our project is to develop a model for computationally creating family trees based on the images in the SSDA. Using each record of baptisms, marriages, and deaths, it may be possible to match the names, locations, and dates to create a story that follows the genealogical progression of the descendants of African slaves in the Iberian New World. Ideally, this tool could be used by a descendant of African slaves who knows (or can infer) the name of an ancestor who appears in these documents, opening up a new chapter of his or her ancestral history that would not have been uncovered otherwise.

We performed this analysis via several steps described below, starting with separating the records from each image. Every image in the SSDA contains at least one record, as well as empty page space and extraneous parts of the image that provide no useful information (such as parts of a table or the fingers of people who scanned the document). It was therefore necessary to isolate the records as blocks of text from the SSDA images to analyze the data efficiently.

Figure 1 shows two example images found in the SSDA [2] that exemplify many difficulties in the quality of historical document data. These images often display extraneous objects, such as fingers and glimpses of the surface that the book is resting on. Likewise, there are differences in lighting and page orientation between these two pages. Moreover, smudging and bleed-through between pages are also clearly evident.

OCR technology is not sufficiently advanced to directly glean character information from such degraded images [16], so another approach must be applied to match names. Our approach treated written names as patterns and matched them with other names that appear similar, thereby addressing the idiosyncrasies of handwriting between different record keepers. Moreover, after matching the records by name, a family tree structure may begin to emerge that helps unlock the ancestral history of millions of people around the world.

The first phase of our approach—separating records from the rest of their images—requires using image segmentation. While segmentation technology has improved

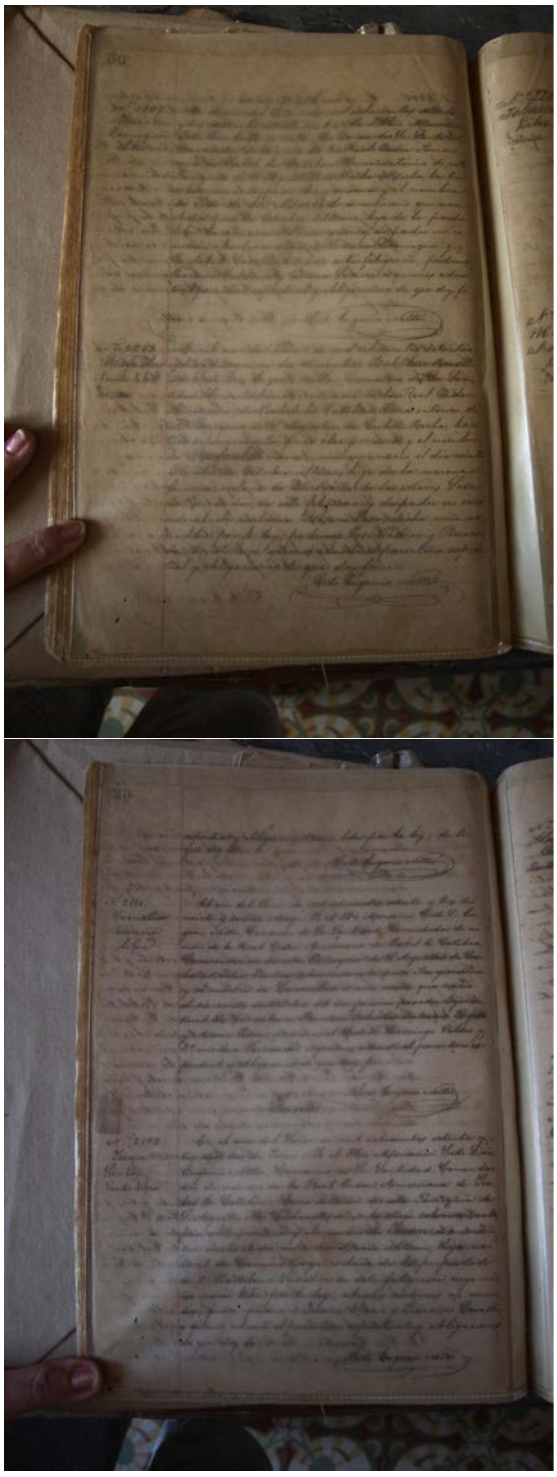

Figure 1: Sample Images from the SSDA.

significantly in recent years, research involving the segmentation of documents has been less expansive. However, the dhSegment [3] model and toolkit demonstrate promising success in this domain using generic computer vision models rather than segmentation-specific document processing tools.

3 Related Work on CNNs

Image processing is a ubiquitous field with extreme variation between different types of images and tasks. A significant number of segmentation methods therefore exist that can be applied for any given task. This section gives an overview of relevant related work, focusing primarily on segmentation using CNN architectures and then discusses how we adapted these methods for use in our segmentation tasks to analyze documents from the SSDA.

One of the first CNNs that appeared in academic literature was AlexNet [5, 17]. AlexNet's main innovation was its more efficient training, which reduced costs and increased the amount of data learned. Some techniques pioneered in AlexNet included using Rectified Linear Units (ReLUs) as activation layers, customized and optimized graphics processing unit (GPU) algorithms for convolutions and training, and pooling outputs together [17].

Building on AlexNet was the Visual Geometry Group (VGG) [23], whose model incorporated the smallest possible convolutions to the earliest layers of the model, allowing for quicker training than its predecessors. Convolutions enabled the creation of feature maps from input data [28], using the smallest possible convolutions to simplify the determination of convolution parameters. Smaller convolutions generally yield more efficient creation of useful feature maps.

Another significant model is GoogLeNet [26], which reduced parameter count and resource usage while training to transition from densely-connected networks to sparsely-connected networks. However, the underlying hardware for modern computations is inefficient when working with sparse calculations. A key contribution of GoogLeNet paper, therefore, was approximating sparse structures with existing dense components, allowing better results with less expensive networks.

A highly significant model is ResNet [10], which provided breakthroughs in training extremely deep networks

by adding skip connections between layers. These skip connections allow the training of the model to ensure that skipped layers perform meaningful tasks, creating a more efficient training process. This approach advanced the field of computer vision significantly.

DenseNet [13] uses the same principle as ResNet to add skip connections between layers. DenseNet, however, adds one layer to every subsequent layer rather than adding connections between every other (or third) layer. Likewise, DenseNet uses fewer convolutional filters per layer due to the large amount of information passed between layers. The analysis of DenseNet in [13] shows that low-level features from early layers are still used by layers closer to the end of the model, which poses questions about how low-level features can be combined with higher-level features.

SqueezeNet [14] is another project that sought AlexNet-level accuracy with their network, but with much less space utilization. The SqueezeNet team recognized that convolutions operating on every input channel use large amounts of space, so they created 1x1 convolutions to squeeze all the input channels into larger convolutions that required fewer parameters. They also used compression techniques, such as Deep Compression [8], to further compact their model while still maintaining accuracy.

4 Our Approach: Applying CNNs for Image Segmentation

Our work in this paper adapts CNNs for segmentation via the U-Net [22] architecture. This architecture leverages the encoder-decoder architecture [15], which uses convolutional layers to encode low-resolution maps of fundamental features in images and subsequently decodes the feature maps to labels for each pixel using upsampling operations, such as pooling and deconvolution [27]. U-Net extends the encoder-decoder architecture by adding skip connections between corresponding downsampling and upsampling layers.

During the decoding process, U-Net combines upsampled data with data that is never fully convolved. This approach maintains information about high-level features in a given image, which allows the model to combine low-level knowledge (such as how to classify pixels) with high-level data (such as where these pixels may be located in the image). The U-Net architecture significantly advanced the field of image segmentation.

A relevant approach specifically focused on the domain of historical document segmentation is called dhSegment [3]. The dhSegment architecture applies common deep learning architectures and standard image processing techniques to perform pixel-wise segmentation via a model similar to U-Net. However, dhSegment uses a ResNet50 [9] model as the encoder and utilizes standard upsampling and concatenation of encoder features as its decoder. A key insight from the dhSegment paper is that a highly successful model for document segmentation can be build via a generic, pre-trained encoder-decoder structure. This model can be trained on a variety of different tasks regarding document segmentation, such as page extraction, layout analysis, and line detection.

The tasks that can be performed on documents is quite expansive. The dhSegment team therefore applied many different types of image processing techniques to further improve the accuracy of their model. Examples of the techniques they applied include threshholding [20] or shape vectorization (which performs a reduction of detected regions into polygonal shapes).

The dhSegment image processing techniques are standard processes that do not require machine learning analyses. Therefore, the task-specific application of post-processing techniques on a general model provide a generalizable tool for document segmentation that requires little training, but instead requires domain knowledge to construct accurate results using simple processing techniques on its output. The success of dhSegment's use of only generic deep learning models as building blocks is impressive, so the rest of this paper evaluates the viability of applying similar generic building blocks to segment historical documents, such as those found in the SSDA.

5 Implementation and Training of the ResNet-based Model

This section describes how we evaluated the ResNet-based dhSegment [3] architecture to other CNN-based models. We first review our experimental dataset, then establish a proof-of-concept and baseline for success, and

finally explain how we implemented and trained the experimental models.

5.1 Overview of the Dataset

The dataset used for our analysis in this paper is based on the SSDA and contains a collection of documents consisting of the baptismal records of people of color from the Iglesia de San Agustín in Ceiba Mocha, Cuba from 1872 to 1892 [2]. Each image in the dataset is a photograph of a page that contains one or more records. As shown in Figure 1, these images also contain extraneous information, such as a hand holding the page flat or a table that the record book is resting on.

Images in the SSDA dataset are labeled such that the two categories of records to extract are different colors from the rest of the image, including the blank page space, table, and other extraneous information. Although this data archive is available publicly on the SSDA website, there are no labels for the data. A dataset of approximately 100 images was created to train and evaluate the performance of a model.

5.2 dhSegment

We applied the pre-trained dhSegment model to form a baseline measure of success by which we can evaluate other models. The dhSegment model implementation and can be found on GitHub [3]. This model is implemented as an application programming interface (API) wrapped around a deep learning model built with TensorFlow and Keras. Since the dhSegment model was trained for document-specific segmentation tasks, its pre-trained weights allow for quick and efficient training.

Using the built-in training method from the dhSegment API, the model can predict the correct pixel value $\sim$92% of the time. Using pixel values, however, can give a skewed measure of performance when large portions of the image are segmented correctly, but are not of interest. The mean Intersection-over-Union (mIoU) is a measure between 0 and 1 representing the ratio of the overlap of predicted and ground truth bounding boxes to the union of the bounding boxes, which is more reflective of successful segmentation. In the present case, the mIoU of the model is $\sim$70.7%. Given the fact that the training dataset contains only $\sim$80 images, this performance is impressive and demonstrates how quickly the dhSegment architecture can learn to analyze ancient documents.

After the initial round of training, our out-of-the-box results were promising. With about 70% mIoU accuracy, however, there was room for improvement. Due to how the dhSegment toolbox is constructed, the ability to look inside the model and make improvements is restricted. Although dhSegment provides a wrapper class created around pure Tensorflow, this wrapper lacked key functionality, such as built-in GPU-optimized data augmentation and the ability to experiment with different loss functions. We therefore constructed other generic models using the FastAI [12] framework and evaluated their performance, as discussed below.

5.3 FastAI

FastAIis an open-source deep learning library and an open API for training and deploying machine learning models [12]. We applied FastAI to provide much of the custom functionality necessary to experiment and augment the capabilities that dhSegment does not provide. For example, FastAI can easily change the loss function of a model during training. Another benefit of FastAI is its ability to add data augmentation when loading the dataset and perform it dynamically along with GPU optimizations. These additions enable relatively quick tuning of a model's hyperparameters that can optimize its performance.

An important feature of FastAI is the function 'unet_learner'. This function allows a user to provide a standard, pre-trained CNN for use as the encoder of a U-Net model, which can then be trained and tested. Likewise, FastAI provides a custom implementation of cross-connections among the encoding and decoding passes of the U-Net so it can operate with any encoder that is provided.

A notable feature of the dhSegment architecture is its ability to combine generic building blocks. The dhSegment team used a ResNet50 architecture for training and evaluation, but their work demonstrated that other generic architectures could work for similar functions. The FastAI library is compatible with any of the models available in the torchvision [18] library, thereby enabling configuration of the dhSegment architecture with any CNN as its encoder. These torchvision models include the ones dis-

cussed in Section 3. Moreover, the torchvision library includes many slightly different alterations of these models. We used FastAI to construct these altered U-Net models and evaluated their performance on the SSDA dataset.

To train and evaluate different generic building blocks in place of a ResNet50 as the encoder of a U-Net, we used the built-in torchvision models available in FastAI, as outlined above. The models we chose were different variations of ResNet, SqueezeNet, DenseNet, VGG, and AlexNet. After the U-Net architecture was created using these pre-trained models as the encoder, we settled on the cross-entropy loss function [25].

In the initial round of training, we performed segmentation into the three classes shown in Figure 2 (main text, column text, and not text), which are representative of the data used in the training and test set. In particular, an image from the SSDA (top) and its corresponding segmentation mask (bottom). The red masks represent the main-body blocks of text, while the green masks represent column blocks of text.

We found that the training process resulted in models that minimized the amount of "not text" that was labeled incorrectly, rather than labeling it correctly as "main text" or "column text." With a larger dataset, we could have used a weighted cross-entropy loss function to account for the imbalance text classes, but with the limited amount of data we elected to combine "main text" and "column text." We therefore performed a binary classification on "text" or "not text" with relatively balanced classes, so a cross-entropy loss function was an appropriate function to minimize.

The model's hyperparameters were selected to be either the default or through cross-validation schemes. We applied FastAI's built-in function to find the optimal learning rates, 'find_lr.' Likewise, each training epoch was performed with the built-in 'fine_tune' function, which includes training defaults specifically used for transfer learning.

The actual training process consisted of training each model between 26 to 30 epochs and evaluating their accuracy on the testing dataset at several checkpoints. This large amount of epochs relative to the amount of training data was performed to train each model to its best performance and then observe how quickly its performance degraded due to overfitting.

To train each model, we began by solely training its

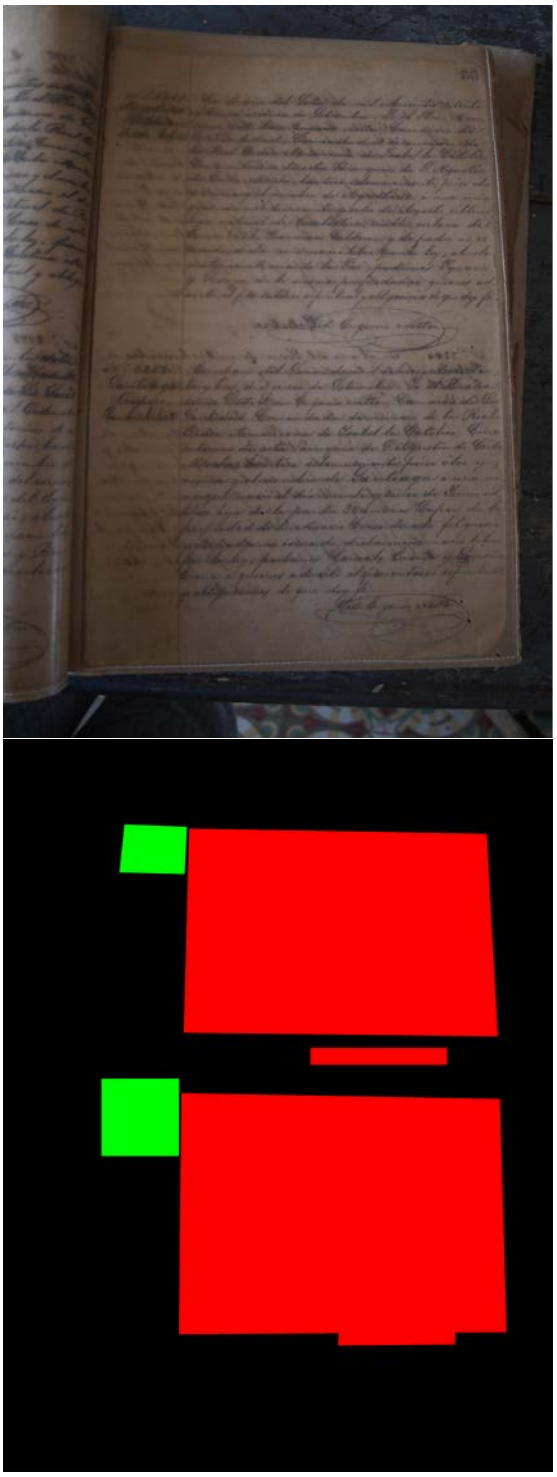

Figure 2: Training Data.

last layer in the first set of epochs. The entire model was then unfrozen and trained. We then progressively froze the earlier layers and continued training until either the models overfit or ∼30 epochs were completed.

6 Analysis of Results

This section presents the results of our evaluation of different base encoders of the U-Net on the given segmentation task. Figure 3 represents the mIoU accuracy over each checkpoint epoch for the different ResNet models evaluated. This accuracy was measured at epoch 5, epoch

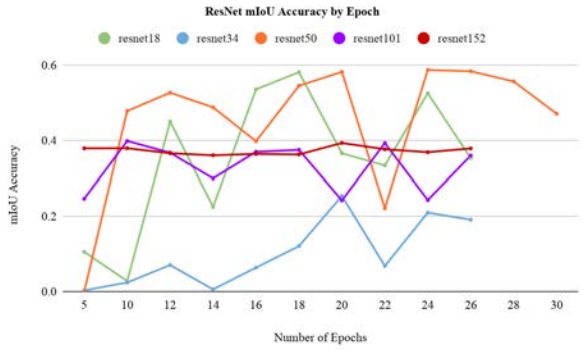

Figure 3: ResNet-encoder Comparison.

10, and every other epoch until the model was determined to overfit sufficiently. The resnet101 base does not overfit, but it never fit in the first place and its relative success was deemed as random based on visual inspection.

Figure 3 shows that the most successful models used resnet50 and resnet18 as their encoders. The model with resnet34 performed the worst by far, whereas the models with resnet101 and resnet152 initially performed well, but did not improve much. This result demonstrates how a CNN can experience tradeoffs as a result of increasing the number of layers.

When there are more layers in a CNN, there are more extracted features as the result of convolutions, but the question remains whether these features can be used efficiently and effectively. As the size of the model grows, the complexity and thus the number of parameters used also increases. Training the models therefore requires changing more parameters, which increases the possibility of overfitting. Overfitting was particularly prominent with resnet101 and resnet152, as they both had high training accuracies, but had essentially random guesses on the testing set.

An interesting aspect of the tradeoff described above is how it operated when going up from 18 layers to 50 layers. Since 18 layers are relatively few, the tendency to overfit was lower, which allowed the model that is based in resnet18 to become accurate quickly without deriving newer features in its layers. The model containing a resnet34 encoder may have had too many layers such that it overfit, but not enough layers to derive any high-level features that may have helped its performance. Finally, the model with a resnet50 encoder had enough parameters such that it may easily overfit, but the added layers gave an extra level of depth that allowed it to work out deeper features and gain accuracy with more training than resnet18 required.

Figure 4 represents the mIoU accuracy for each of the SqueezeNet and DenseNet models evaluated. This accu-

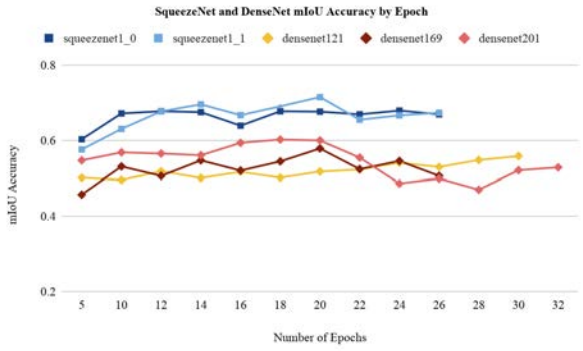

Figure 4: SqueezeNet-encoder and DenseNet-encoder Comparison.

racy was measured at epochs 5, 10, and every other epoch thereafter. After learning what the targets for segmentation were, the models did not improve or worsen significantly over many epochs.

Figure 4 shows how both variations with SqueezeNet encoders had the best accuracy of all the out-of-the-box

models available on FastAI. DenseNet encoders also performed extremely well. Both these base encoder architectures created models that performed accurately without much training, but did not significantly increase their levels of accuracy or begin to overfit after excessive training. One explanation for why these models perform better than ResNet is that they both have significantly fewer parameters than ResNet, which is consistent with the same logic that allows the model based on the resnet18 encoder to perform well.

Figure 5 represents the mIoU accuracies achieved for the VGG and AlexNet models throughout their training. AlexNet had high accuracy in the beginning, but

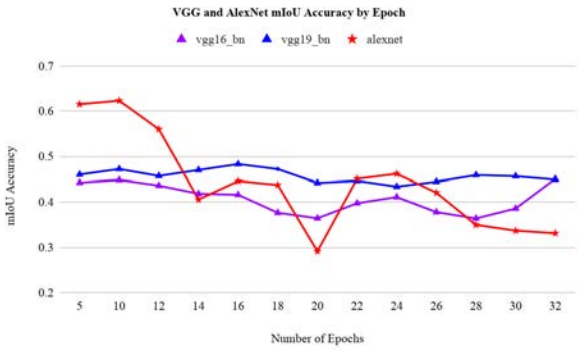

Figure 5: VGG-encoder and AlexNet-encoder Comparison.

dropped off after training. This result did not occur due to chance, as this happened multiple times when training from scratch.

Figure 5 demonstrates how AlexNet required little training to perform well, while VGG did not change its performance even after training extensively. AlexNet is a successful—yet relatively early—CNN that is composed of few layers compared to the CNNs that were created since. It is much smaller and has fewer parameters, which helps explain how it quickly identified the important features in training, but also how further training allowed for overfitting since it lacked the depth required to make complex features.

Figure 6 displays the same data between the best of each type of model. The testing accuracy for the best

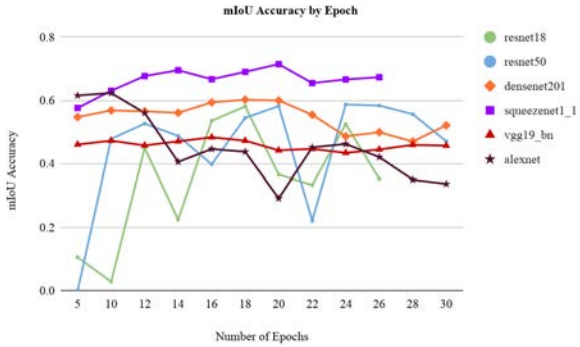

Figure 6: Inter-model Comparison.

from each model type. Two types of ResNet models were included since they performed quite similarly, which is notable and will be discussed below. This comparison of different model types allowed easy visualization with relative accuracy and variability.

Figure 6 provides an overview of how all the types of models compare against each other. These results show how ResNet-based encoders provide much more variability in the model. The model may be basing its decision-making on features that provide different results when slightly altered. Another notable result shown in Figure 6 is how several of the tested CNN encoders create models that do not perform much better or worse after training than before.

Our finding that SqueezeNet and DenseNet encoders performed better than the other models for a majority of the training and evaluation process indicates the need for further research into these architectures and their applicability in document analysis. It is noteworthy that the SqueezeNet output shown on the in Figure 7 appears more block-based than the ResNet, which helps explain how SqueezeNet can encapsulate blocks of text with success. This figure shows the real (left) and predicted (right) output of a test set image for both squeezenet1_1 (top) and resnet50 (bottom) encoder-based models. By visual inspection, the SqueezeNet based model encapsulates the logic of where text blocks may occur better than the ResNet-based model.

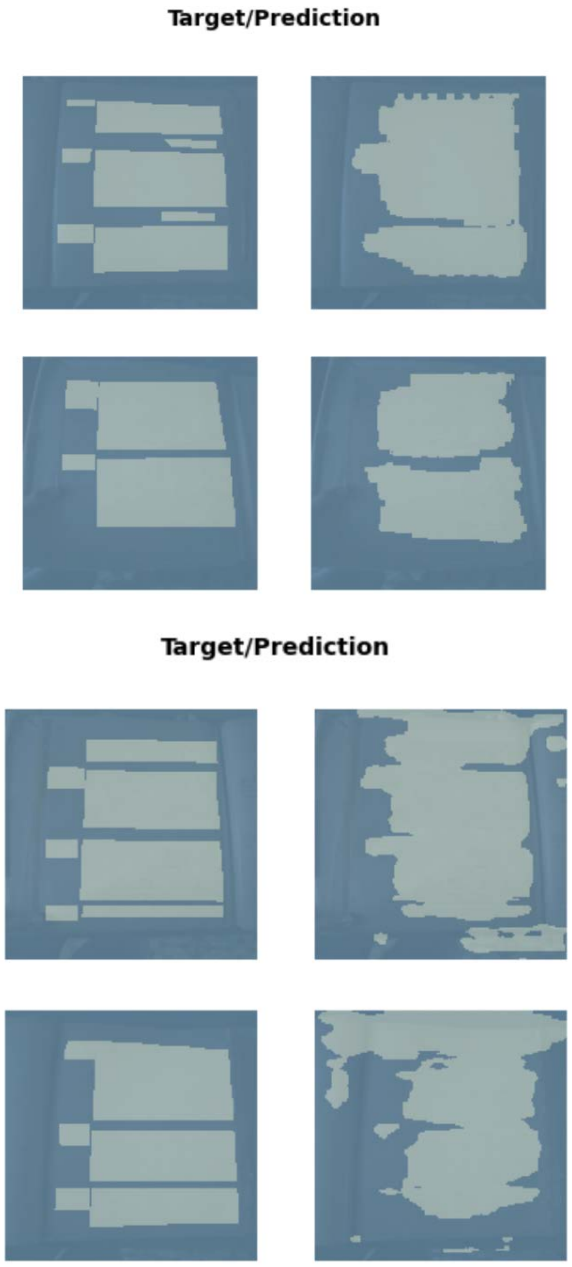

Figure 7: SqueezeNet and ResNet Outputs

7 Concluding Remarks

Conventional pre-trained CNNs are not well suited to perform segmentation on historical documents, such as the Slave Societies Digital Archives (SSDA) used as the basis for our research presented in this paper. Relatively little literature examines document segmentation since the rise in popularity of CNNs. Some research, however, is beginning to push the envelope.

For example, the dhSegment paper and library provide a promising approach and toolkit for segmentating historical documents. The dhSegment authors demonstrated the feasibility of general computer vision models and standard post-processing techniques by inserting a ResNet50 encoder into the U-Net architecture and outperforming specifically dedicated systems built for document segmentation. Our research expanded upon their approach by training and evaluating other general-use computer vision models as the encoders in the U-Net architecture.

The key lessons learned we gleaned from this research are summarized below:

- **Deep learning may outperform mature, classical methods of document analysis.** The lack of ubiquity of a single or subset of classical methods used for image-based document analysis demonstrates that the existing tools are not robust enough. Deep learning is a relatively new technique, but it is already making significant progress in creating successful document analysis [5].

- **Generic deep learning techniques—rather than specialized document analysis systems—are successful in historical document segmentation.** The advent of the dhSegment toolbox [3] showed that the combination of generic deep learning building blocks, ResNet, and U-Net architectures, yielded promising results demonstrating that a CNN can label historical documents sufficiently well.

- **Other unspecialized deep learning building blocks have the potential to improve on dhSegment's original architecture.** Our results showed that other generic CNN-based architectures, specifically using SqueezeNet [14] and DenseNet [13], outperformed ResNet50 on our specific dataset. This result is not definitive due to the limited size and scope

of the data used, but it is nonetheless an interesting outcome.

- **With a relatively small amount of data, we were able to train and evaluate several CNNs**, such as other ResNets, SqueezeNet, DenseNet, and more. Our initial hypothesis that a ResNet50 encoder would perform the best on segmentation tasks rather than other generic building blocks was not supported by our empirical evidence. Due to the lack of diversity within the dataset and the small number of images analyzed, our results are not conclusive that any of these given models work better than ResNet. However, we demonstrate that other models like SqueezeNet and DenseNet perform better on our specialized dataset and should be considered targets for further research in the context of document segmentation.
- **Due to confounding issues (such as the extraneous objects and page bleed-through shown in Figure 1), more research must be conducted to advance this domain of analysis**. While this paper does not provide a comprehensive model that completely solves the first phase of the eventual family tree problem, it does provide the foundation for future attempts of this problem and many others that lie adjacent to it. In particular, our results empirically evaluate potential analyses that help to further the success of historical document segmentation.

Our future work consists of exploring the performance of these architectures on larger datasets and incorporating them into a toolbox with the post-processing techniques mentioned by the dhSegment team. Likewise, we are exploring the intricacies of the training process concerning the variability seen in training the ResNet models and the lack thereof within the training of the DenseNet models and others. Finally, the use of the transformer-based multidimensional long-short-term-memory [24] (which is another type of artificial intelligence model) is a promising technique for document analysis that we are exploring.

Additional Information

Parts of this chapter were previously published in the Master's thesis by the same author: Evan Segaul. "Evaluation of Generic Deep Learning Building Blocks for Segmentation of 19th Century Documents," 2021, [Unpublished Master's thesis]. Vanderbilt University. Available from: https://ir.vanderbilt.edu/handle/1803/16673.

References

[1] Slave societies digital archive. https://slavesocieties.org/home. Accessed: 2021-2-10. 1, 2

[2] Libro 7 de bautismos de pardos y morenos, 1872-1892, June 2020. 2, 5

[3] S Ares Oliveira, B Seguin, and F Kaplan. dhsegment: A generic Deep-Learning approach for document segmentation. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 7–12, Aug. 2018. 1, 2, 3, 4, 5, 9

[4] S Belongie, C Carson, H Greenspan, and J Malik. Color- and texture-based image segmentation using EM and its application to content-based image retrieval. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 675–682, Jan. 1998. 1

[5] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. Apr. 2017. 3, 9

[6] Spyros Gidaris and Nikos Komodakis. Object detection via a multi-region and semantic Segmentation-Aware CNN model, 2015. 1

[7] Yanhui Guo and Amira S Ashour. 11 - neutrosophic sets in dermoscopic medical image segmentation. In Yanhui Guo and Amira S Ashour, editors, *Neutrosophic Set in Medical Image Analysis*, pages 229–243. Academic Press, Jan. 2019. 1

[8] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. Oct. 2015. 4

[9] S R Hashemi, S S Mohseni Salehi, D Erdogmus, S P Prabhu, S K Warfield, and A Gholipour. Asymmetric loss functions and deep Densely-Connected networks for Highly-Imbalanced medical image segmentation: Application to multiple sclerosis lesion detection. *IEEE Access*, 7:1721–1735, 2019. 1, 4

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. Dec. 2015. 3

[11] Zhaofeng He, Tieniu Tan, Zhenan Sun, and Xianchao Qiu. Toward accurate and fast iris segmentation for iris biometrics. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(9):1670–1684, Sept. 2009. 1

[12] Jeremy Howard and Sylvain Gugger. Fastai: A layered API for deep learning. *Information*, 11(2):108, Feb. 2020. 5

[13] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. Aug. 2016. 4, 9

[14] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and ¡0.5MB model size. Feb. 2016. 4, 9

[15] Yuzhu Ji, Haijun Zhang, Zhao Zhang, and Ming Liu. Cnn-based encoder-decoder networks for salient object detection: A comprehensive review and recent advances. *Information Sciences*, 546:835–857, 2021. 4

[16] A S Kavitha, P Shivakumara, G H Kumar, and Tong Lu. Text segmentation in degraded historical document images. *Egyptian Informatics Journal*, 17(2):189–197, July 2016. 1, 2

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.*, 25:1097–1105, 2012. 3

[18] Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM international conference on Multimedia*, MM '10, pages 1485–1488, New York, NY, USA, Oct. 2010. Association for Computing Machinery. 5

[19] M Nilsson, A H Herlin, H Ardö, O Guzhva, K Åström, and C Bergsten. Development of automatic surveillance of animal behaviour and welfare using image analysis and machine learned segmentation technique. *Animal*, 9(11):1859–1865, Nov. 2015. 1

[20] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, Jan 1979. 4

[21] D L Pham, C Xu, and J L Prince. Current methods in medical image segmentation. *Annu. Rev. Biomed. Eng.*, 2:315–337, 2000. 1

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. May 2015. 4

[23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for Large-Scale image recognition. Sept. 2014. 3

[24] Marijn F Stollenga, Wonmin Byeon, Marcus Liwicki, and Juergen Schmidhuber. Parallel Multi-Dimensional LSTM, with application to fast biomedical volumetric image segmentation. June 2015. 10

[25] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 240–248. Springer International Publishing, 2017. 6

[26] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. Sept. 2014. 3

[27] Saeid Asgari Taghanaki, Kumar Abhishek, Joseph Paul Cohen, Julien Cohen-Adad, and Ghassan Hamarneh. Deep semantic segmentation of natural and medical images: a review, 2021. 4

[28] J Wang, J Lin, and Z Wang. Efficient convolution architectures for convolutional neural network. In *2016 8th International Conference on Wireless Communications Signal Processing (WCSP)*, pages 1–5, Oct. 2016. 3

[29] Jules White Douglas C. Schmidt Zhongwei Teng, Quchen Fu. Sketch2vis: Generating data visualizations from hand-drawn sketches with deep learning. In *International Conference on Machine Learning and Applications (ICMLA)*, 2021. 1