Puzzle Piece Segmentation using Gaussian Mixture Model and U-Net

Thishen Packirisamy - 1839434
University of the Witwatersrand
School of Computer Science and Applied Mathematics

Kavilan Nair - 1076342
University of the Witwatersrand
School of Computer Science and Applied Mathematics

***Abstract*—This report investigates puzzle piece segmentation using both traditional statistical and deep learning computer vision techniques. A Gaussian Mixture Model (GMM) and U-Net model were implemented and their performance compared on a 48 piece puzzle dataset. A detailed overview of both methods are presented along with the implementation. The U-Net models outperformed the GMM model across all evaluation metrics. The GMM performed with a mean Intersection over Union (mIoU) of 0.953 and the U-Net model trained on the original dataset achieved a mIoU of 0.969. The U-Net model trained on the augmented dataset performed the best with a mIoU of 0.982.**

I. Introduction

Solving jigsaw puzzles manually can be extremely difficult depending on the number of pieces and the complexity of the puzzle image. The process of solving a jigsaw puzzle can be automated by computers which leverage clever computer vision algorithms. The first step in an automated jigsaw puzzle solver is to first obtain images of each individual puzzle piece and generate a mask from the image. The puzzle piece masks can then be used by other algorithms to determine how they fit together. This paper focuses on the first part of the jigsaw puzzle solver which is the puzzle piece segmentation problem.

The puzzle piece segmentation task can be seen as a semantic image segmentation problem where a model needs to predict whether each pixel in an image belongs to the puzzle piece or the background. Two approaches are taken to solve the binary segmentation task in this paper, a tradition statistical learning approach which makes use of a Gaussian Mixture Model (GMM) and a deep learning approach which makes use of the deep convolutional U-Net model architecture.

This report includes an overview of both the GMM and U-Net models, in addition to the implementation details and evaluation of both approaches.

II. Background

A. GMM Overview

A Gaussian Mixture Model (GMM) is a probability distribution that combines K different Gaussian distributions into a single model. This is beneficial over a single Gaussian distribution as it allows the model to pick on a high number of distinct phenomena in the data. GMMs overcome the common problems of unimodality and robustness by allowing multiple Gaussians to fit certain phenomena within the data. GMM formula:

$$Pr(X|\theta) = \sum_{k=1}^{K} \lambda_k Norm_X[\mu_k, \Sigma_k]$$

Making use of use of a hidden categorical variable h we can express the GMM as a marginalization:

$$Pr(X|h, \theta) = Norm_X[\mu_h, \sigma_h]$$

$$Pr(h|\theta) = Cat_h[\lambda]$$

using this formulation of the distribution we can sample from the categorical prior $Pr(h)$ first and subsequently, we can sample from the joint distribution $Pr(A|h)$ using the prior. This, therefore allows us to interact with the individual Gaussian distributions within the overall mixture. Using the marginalization we can iteratively fit the model to a random variable X using Expectation Maximization.

B. U-Net Overview

Semantic image segmentation is a task that involves assigning a label to every pixel contained in an image. The puzzle piece segmentation problem is a two class semantic image segmentation problem where the first class represents the puzzle piece and the second class represents the background. Deep learning techniques have provided state-of-the-art performance in computer vision problems such as image classification and semantic image segmentation. The increase in compute power, large-scale quality labelled datasets and clever innovations such as deep Convolutional Neural Networks (CNNs) have all contributed to the recent success [1].

There are several advanced image segmentation models such as DeepLab v3 and MaskR-CNN [2], [3]. Whilst these models perform extremely well, the puzzle piece segmentation problem is relatively simple meaning a simpler model should be able to perform the task with high accuracy. U-Net, a deep convolutional neural network that was was initially developed for the application of biomedical image segmentation was shown to perform well on a limited dataset [4]. These properties make it suitable for the puzzle piece segmentation task and was implemented in this paper.

The U-Net architecture is shown in figure 1 and consists of two main components, a contracting path and an expansive path. The contracting path component resembles a traditional

CNN which consists of convolutional layers followed by a ReLU layer and max pooling layer. The expansive path contain upsampling layers followed by up-convolution layers and also concatenate the corresponding feature map from the contracting path. These concatenations are known as skip connections.

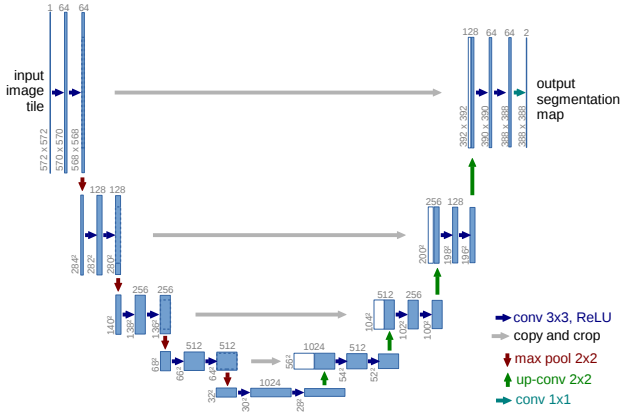

Fig. 1. U-Net Architecture [4]

III. Methodology

A. GMM Implementation

1) Data: The dataset contains 48 images. 8 images, that correspond to the U-Net test set, were separated to create the test set for the GMM. Of the 40 remaining images, a 5-fold cross-validation set was created. The images in the original dataset are relatively large for the problem space with a resolution of 1024x768. It was decided that the images would be scaled by a factor of 0.25.

2) Features: In order to create the training set, some feature engineering was experimented with. The following feature sets were tested: RGB pixel values, RGB pixel values with Difference of Gaussian filtered image and HSV pixel values. It was found that the HSV pixel values outperformed the other feature sets. This is likely due to the fact that the HSV colourspace is more robust to external changes in lighting than RGB.

3) Hyper-parameter optimisation: The 5 Fold cross-validation set was used to optimize the hyper-parameters of the GMM. A hyper-parameter search was performed on each fold by holding out the fold as a validation set and training on the rest of the data. Table I shows the hyper-parameters search space and best values. The evaluation metrics were taken for each fold and averaged to find the best hyperparameters.

TABLE I
Hyper-parameters

Hyper-parameter	Search space	GMM
Feature Set	[RGB, RGB+DoG, HSV]	HSV
Num Gaussians Foreground	[2, 3, 4]	3
Num Gaussians Background	[2, 3, 4]	3

4) Training: An Expectation-Maximization (EM) approach was taken to find the means and variances of Gaussians within the GMM. Three Gaussian components are used for both the foreground classifier and the background classifier. The HSV feature set of the 40 train images is used train the classifier. The maximum number of iterations of EM is set to 100 with a minimum of 10. In some cases during training the distribution locks on to a single point in the space. This causes the covariance to be singular which leads to an infinite likelihood. To combat this, a diagonal matrix the size of the covariance matrix is added to the covariance. The matrix contains very small values along the diagonal and adding this to the covariance matrix ensures the determinant is not 0 and therefore the matrix is non-singular.

B. U-Net Implementation

1) Architecture: The U-Net model was implemented using the TensorFlow Python machine learning library. Deep learning requires large amounts of data and compute power to train accurate models. The puzzle-piece dataset only contains 48 images which is an extremely small dataset in the context of deep learning and when considering the number of parameters that require tuning in the U-Net model. It is common practice to initialise the weights of the model using another models weights that have been pre-trained on a large dataset such as ImageNet [5]. Even though ImageNet is not a puzzle piece dataset, the model learns meaningful representations and transfer learning can be used to fine tune the model for the task of puzzle piece segmentation with the limited dataset. The U-Net model weights were initialized using the weights from a VGG16 model that was pre-trained on ImageNet [6]. Initializing the weights using a pre-trained network allows for the model to converge much quicker than if it was starting from randomly initialized weights.

2) Loss Function: Supervised learning approaches in machine learning require a loss function to quantify how well a model is performing and is required to optimize and improve the model. The task of classifying each pixel in an image as either puzzle piece (foreground) or background is a binary classification problem. The Binary Cross Entropy (BCE) loss is often used in binary classification problems and is shown in equation 1 where y is the target value and p is the predicted probability. The sigmoid activation function is required to be used at the last layer.

$$BCE = -(ylog(p) + (1-y)log(1-p)) \quad (1)$$

3) Data Resizing and Splitting: The dataset contains images with a resolution of 1024x768, which is a relatively high resolution for the puzzle piece image data. The U-Net model contains a large number of parameters and requires a large amount of GPU memory when training. High resolution images therefore restrict the batch size that can be used. It was decided that both the input image and the masks should be scaled by a factor of 0.25. This enables larger batch sizes to be used and also speeds up the training.

K-Fold cross validation is not used for the U-Net model as it will take an extremely long time to train an individual model across each fold. Instead, the traditional train, validation and test data split is applied. The test set consists of the same unseen test images used in the GMM implementation for consistency. This allows for a fair and consistent comparison of evaluation metrics between the two approaches on the test data set.

4) Data Augmentation: Data augmentation is a technique where various augmentations are applied to image data in order to create new training examples and add variety to the dataset. Data augmentation can help improve the models performance on unseen data as it is trained on a more diverse dataset. Data augmentation for image data can be categorized as either spatial augmentations or pixel augmentations. Spatial augmentations include techniques such as flipping and rotating images to produce new instances of the image data. Pixel augmentation involves creating new training examples that have the pixel value intensity varied such as the brightness or hue of the image. The following augmentations were applied and added to the training dataset:

- Flip image horizontally
- Flip image vertically
- Adjust brightness by random factor
- Adjust contrast by random factor
- Adjust saturation by random factor

After the data augmentations were applied, the training dataset consists of 192 images and 48 validation images. Models trained on the original scaled dataset and augmented dataset were trained and are compared in the results section.

5) Hyper-parameter optimization: Training the U-Net model requires a few different hyperparameters to be set and tuned to achieve the best possible performance. The network architecture was not altered between runs as we implemented U-Net with VGG16 weights. The Optuna Python hyper-parameter optimization library was used to search the hyper-parameter space for the best possible combination [7]. The hyper-parameters that resulted in the best validation accuracy were then used to train the final models.

TABLE II
HYPER-PARAMETERS

Hyper-parameter	**Search space**	**Model 1**	**Model 2**
Learning rate	[0.1, 0.01, 0.001]	0.001	0.001
Epochs	[50, 100]	50	50
Batch size	[4, 8, 16]	8	4

C. Evaluation Metrics

There are several metrics one can use when evaluating the performance of a semantic image segmentation model. Metrics provide a numeric value that can then be used to compare with other models using the same metrics. This makes it possible to determine which of the models perform the best. Both GMM and U-Net models were evaluated using the metrics outlined in this section.

1) Pixel Accuracy: Pixel accuracy is defined as the percentage of pixels that are classified correctly and is defined by equation 2, where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

The pixel accuracy metric does have a flaw, if there is a large class imbalance a high pixel accuracy does not mean that the model is performing particulary well. In the context of puzzle piece segmentation, if there puzzle piece mask only uses 10% of all pixels in an image and the model predicts all pixels as background, the pixel accuracy will be 90% which seems high but the model has completely failed to segment the puzzle piece.

2) Mean Intersection over Union: The Intersection over Union (IoU) evaluation metric is popular for semantic image segmentation tasks [8]. IoU is calculated using equation 3, where A is the predicted region and B is the ground truth region. If the predicted and actual region are identical which is the ideal scenario, the metric evaluates to 1 and if no overlap exists, it evaluates to 0. The mean IoU metric is the averaged IoU for each class present.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{3}$$

3) Dice Similarity Coefficient: The Dice similarity coefficient is a metric that measures the similarity between two sets of data and is also commonly used in semantic image segmentation. It is calculated using equation 4.

$$DSC = \frac{2TP}{2TP + FP + FN} \tag{4}$$

4) RoC-AUC: The Receiver Operating Characteristic curve (ROC) is a plot which visualizes the performance of a binary classifier. The Area Under the Curve of the RoC (RoC-AUC) is a single value which can be used as a performance metric and is used to evaluate the models presented in this paper.

IV. Results and Evaluation

The best performing models from the hyper-parameter optimization procedures for both the GMM and U-Net models are presented and compared in this section. Multiple evaluation metrics are calculated for each model on an identical unseen test dataset. A single GMM model is compared to the U-Net model fine-tuned on the original dataset and another U-Net model which has been trained on the dataset with image augmentations.

Table III contains all the performance metrics for the three different models when evaluated on the unseen test dataset.

All models are able to perform extremely well with all models performing with similar results. The GMM performed slightly worse than the two U-Net models across all four recorded metrics. The GMM performed with an mIoU of 0.953 compared to an mIoU of 0.969 and 0.982 for the two U-Net models.

TABLE III
EVALUATION RESULTS

	GMM	U-Net	U-Net Aug
Accuracy	0.987	0.989	0.994
mIoU	0.953	0.969	0.982
Dice	0.976	0.984	0.991
RoC AUC	0.981	0.992	0.993

Although handcrafted features were used in the GMM implementation and previous models, the model still performed worse than Unet which took raw RGB data. This may be due to the fact that the CNN model itself learns features from the data. In this regard, it can be seen that the CNN model learns better filters and features that fit the problem space than we can create ourselves.

The U-Net model that was trained on the augmented dataset performed slightly better than the U-Net model that was trained on the original data. This implies that the data augmentations that were applied helped provide valuable new images that allow the model to generalize better on the unseen test data.

Previously a more simple single Gaussian model with a Bernoulli prior was used to segment the puzzle pieces. This model vastly outperforms the GMM in terms of training time as it trains around 350 times faster than the GMM. The inference is also faster as it runs at about 2 times the speed of the GMM. The GMM however performs better having an accuracy of 0.987 compared to the 0.955 of the previous single Gaussian model.

An attempt was made to analyse what aspect of the image each Gaussian in the GMM was responding to. This was done by running the background and foreground prediction using only a single Gaussian from the GMM on the test data set. The results of this did not reveal what underlying subclasses of the data the model was actually learning. Further investigation and experiments should be done to determine what exactly was being learnt.

The GMM predicted test masks are shown in figure 2. It can be seen that although the GMM achieves a high accuracy there are obvious errors and noise in the predicted masks.

The U-Net predicted test data masks are shown in figure 3. Even though the model achieved extremely high evaluation metrics, it can be seen that it does not perform perfectly. There are regions outside of the actual puzzle piece that it classifies as foreground. There are also some edges that the model is unable to predict extremely well.

The puzzle piece masks shown in figure 4 were predicted by the U-Net model that was trained with augmented data. The masks do look slightly better than the masks generated by the first U-Net model as it does not contain as many regions outside the mask that were classified as foreground. There are however small regions contained within the puzzle piece mask that it predicts as background. The predictions around the edges of the puzzle pieces are more accurate which is important in the context of using these masks to solve the overall puzzle.

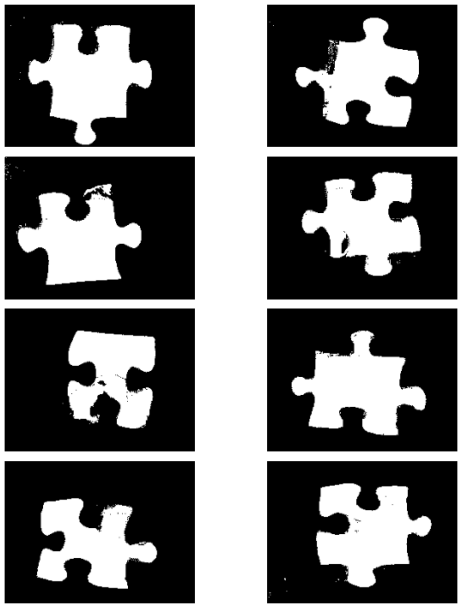

Fig. 2. GMM predictions

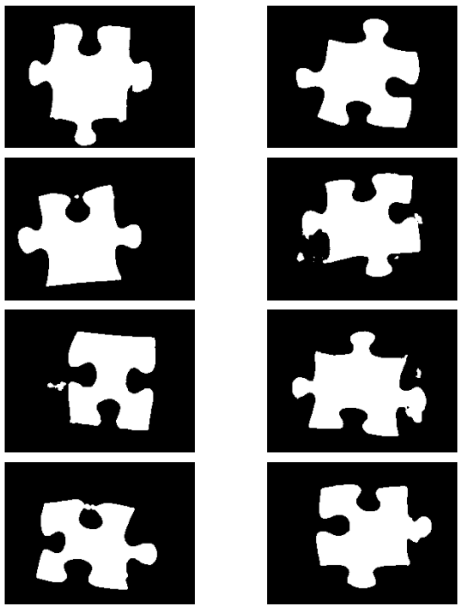

Fig. 3. U-Net predictions

V. CONCLUSION

This report details the implementation and results obtained using a GMM and U-Net model on a puzzle piece segmentation problem. The deep-learning U-Net model outperformed the GMM model on all evaluation metrics. The U-Net model trained on the original dataset achieved a mIoU of 0.969 and the U-Net model trained on the dataset with augmentations achieved an moU of 0.982 whilst the GMM performed with a mIoU of 0.953. The GMM does require less compute to train than the U-Net models but after evaluating the predicted

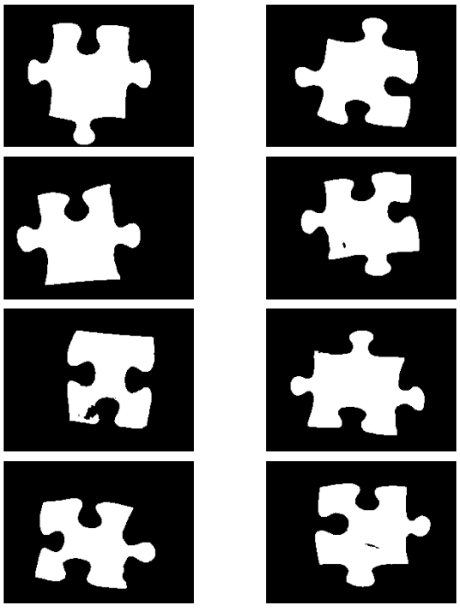

Fig. 4. U-Net Aug predictions

masks on the test dataset, it can be seen that the U-Net models produce a superior mask. It is recommended that the U-Net model trained with data augmentation is used to solve this problem.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[2] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017.

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," 2018.

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[7] Optuna. [Online]. Available: https://optuna.org/

[8] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 658–666.

APPENDIX

TABLE IV
WORK DISTRIBUTION

Component	Thishen Packirisamy	Kavilan Nair
GMM Implementation	✓	
U-Net Implementation		✓
GMM Write up	✓	
U-Net Write up		✓
General report	✓	✓