Oneshot Pokémon Identification Using Markov Random Fields

Nathan Morrison, Logan Padon

May 10, 2019

1 Problem Description

While neural learning is making image identification easier than ever, it requires a ton of data to generate good results. For many image identification problems, however, data is limited and it's impossible to create the database necessary to support a CNN. Oneshot image identification is a subset of that problem, where a model is trained on only one image per class.

Oneshot learning is difficult, but by trying to identify Pokémon, we can make it comparatively easy. Pokémon are designed to be easily identifiable by children, and often have bright, contrasting colors and unique shapes. Unlike a naturally occurring animal, they rarely have pattern variations or color changes.

Our dataset [1] consists of the in-game sprite for each of the first 720 Pokémon. Due to the length of time it takes to calculate the Z value, we will be limiting ourselves to the 151 Pokémon from the original games. These sprites will be the data we use to train the model. Then, we will use images of artwork from the trading card game (TCG) to test and validate the model. The TCG is the largest source of art for Pokémon outside of official sprites, and it provides an interesting challenge for the model, as it often depicts the Pokémon in different art styles, such as 3D models or watercolor art, and in different poses. While some Pokémon have alternate forms, and all Pokémon have a "shiny" form with alternate colors, we have pruned the TCG database of these forms for the sake of simplicity.

2 Previous Work

The creators of the dataset used a softmax CNN model and achieved 30% accuracy [1].

Koch, Zemel, and Salakhutdinov used Siamese Neural Networks for oneshot learning on omniglot data [2].

Kass, Witkin, and Terzopoulos outlined the snakes algorithm, which segments images by contours. Their work forms the basis of our approach to background segmentation [3].

3 Approach

Background segmentation. While we did not use background segmentation due to computing time constraints, we will detail our approach to the problem here. If we could effectively subtract the backgrounds from the image, we would have color data that more closely corresponds to the Pokémon, which could improve accuracy significantly. We used active contour segmentation to segment out the background [4]. This approach segments the image by minimizing an energy value that depends on both the image and the length and smoothness of the contour lines. Unfortunately, the model was ineffective on our images. Our theory is that, due to the art style of the images, contours are more difficult to find. Especially on watercolor images, finding a real "edge" between the Pokémon and the image is very difficult, and the results were so poor that the we clearly could not use them in the final results. In researching the problem, we found that the best answers would all require a lot of annotated data [5], so the idea was scrapped.

Color identification. We've found that there are rarely more than 5 unique colors for a given Pokémon, so we used K-means clustering to find the 5 most common colors for each of the original sprites. From that, we calculated the normalizing constant Z using the following algorithm:

Data: The 5 means for a given image, M
Result: Z, the normalizing constant
initialization;
foreach $c \in AllPossibleColors$ **do**
  foreach $m \in M$ **do**
    $Z \mathrel{+}= \frac{1}{euclideanDistance(c,m)}$;
  end
end

Algorithm 1: Solving for Z for a given image

The Euclidean distance allows us to calculate the differences between the colors. By summing the inverse of those differences, we are able to find the normalizing constant. From there, we calculate the $P(imageColors|classColors)$ using the following algorithm:

Data: The 5 means for a given image, M, and the 5 means for the class, $classColors$
Result: $P(imageColors|classColors)$, the probability of the image being in class, given the class's colors
initialization;
foreach $m \in M$ **do**
  foreach $c \in classColors$ **do**
    $d = \min(d, euclideanDistance(c, m))$
  end
  $p \mathrel{\times}= \frac{1}{Z} \times \frac{1}{d}$
end

Algorithm 2: Solving for $P(imageColors|classColors)$ for a given image

We can use algorithm 2 to create one list of probabilities for each class. Then, we can take the class with the maximum probability as the class for the model, and compare it against the actual class to calculate our model's accuracy.

4 Experiments

We tested the model on 937 TCG images, but only correctly identified 25. While that is only an accuracy of 2.6668%, it is better than random, suggesting that color was a good identifying feature, and that with better image preprocessing, we could achieve a better result.

5 Conclusion

We have presented an above-random approach to the one-shot Pokémon identification problem using K-means color clustering. Our model suggests that color clustering is a step in the right direction, but by itself is insufficient for solving this problem. One major issue for our model is that calculating the Z value for a given training image can take up to half an hour on normal computers, and even with advanced resources it still takes a few minutes. Our work could be improved by adding a better layer of background segmentation to the test image, but due to the constraints of our data and of computing time, we were not able to effectively segment out the background of our data.

References

[1] One-Shot-Pokemon Images database,

https://www.kaggle.com/aaronyin/oneshotpokemon

[2] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese Neural Networks for One-shot Image Recognition."

[3] Snakes: Active contour models. Kass, M.; Witkin, A.; Terzopoulos, D. International Journal of Computer Vision 1 (4): 321 (1988). DOI:10.1007/BF00133570

[4] P. Pandey, "Image Segmentation using Python's scikit-image module," Towards Data Science, 15-Feb-2019. [Online]. Available: https://towardsdatascience.com/image-segmentation-using-pythons-scikit-image-module-533a61ecc980. [Accessed: 27-April-2019].

[5] T. Bouwmans, S. Javed, M. Sultana, and S. K. Jung, "Deep Neural Network Concepts for Background Subtraction: A Systematic Review and Comparative Evaluation."