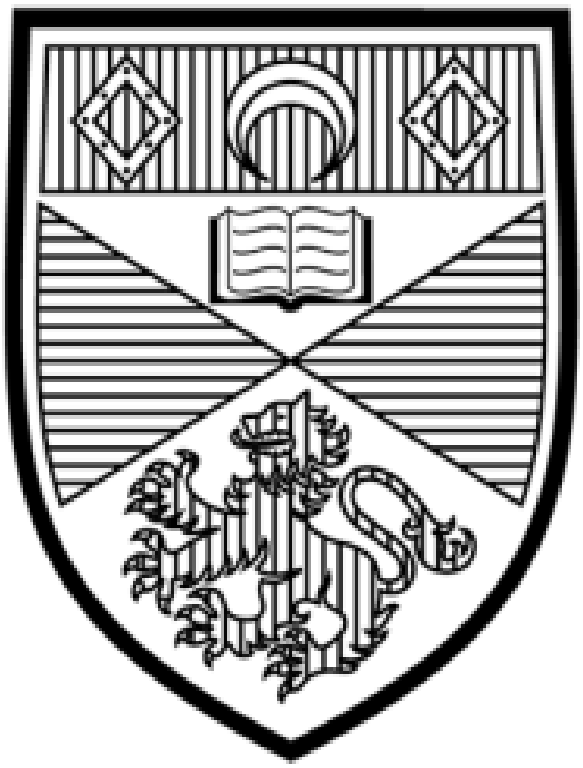
140002133 George Alexander 9 April 2018

CS4098 Minor Software Project

Learning to Generate Line Drawings from Examples



University of
St Andrews

Supervisor: Dr Kasim Terzic

Abstract

This project explores the problem of using Artificial Intelligence techniques to enable computers to create original line drawings of objects. The current range of image creation algorithms is extremely limited. Current approaches either generate unrealistic images or cannot address tasks that require important properties of images like large shapes. This project takes line drawings of objects, and by splitting the complete contour around that shape into fragments, where a fragment is a line that represents some proper part of the whole contour, and comparing fragments, be able to probabilistically recombine them into new images that can themselves be convincing. The relationship between parts of an image are estimated to allow the create new images (in this case line drawings) through a variety of methods that have a more realistic and plausible look than other algorithms. This is of inherent interest as a demonstration of some limited form of artificial creativity but could also be used as an interesting illustration of the statistical techniques involved, or be a starting point for more advanced algorithms to address image generation.

Declaration:

"I declare that the material submitted for assessment is my own work except where credit is explicitly given to others by citation or acknowledgement. This work was performed during the current academic year except where otherwise stated. The main text of this project report is 11,200 words long, including project specification and plan. In submitting this project report to the University of St Andrews, I give permission for it to be made available for use in accordance with the regulations of the University Library. I also give permission for the report to be made available on the Web, for this work to be used in research within the University of St Andrews, and for any software to be released on an open source basis. I retain the copyright in this work, and ownership of any resulting intellectual property."

Contents

1. General Introduction

The aim of this project is to explore the problem of using Artificial Intelligence techniques to enable computers to create original line drawings of objects. The current range of image creation algorithms is extremely limited. There are many techniques for image interpolation and Google DeepMind have attempted to turn their Generative Adversarial Networks (GAN) to the general problem of image generation (Mordvintsev, Olah, Tyka, 2015). However, these approaches either generate unrealistic images or cannot address tasks that require important features of images like large shapes. The approach of this project is to take line drawings of objects, and by splitting the complete contour around that shape into fragments, where a fragment is a line that represents some proper part of the whole contour, and comparing fragments, be able to probabilistically recombine them into new images that can themselves be convincing.

This is a shape

This is a fragment of the same shape

If we can adequately model the relationship between parts of an image there is the possibility that we may be able to create new images (in this case line drawings) that have a more realistic and plausible look than other algorithms. This is of inherent interest as a demonstration of some limited form of artificial creativity but could also be used as an interesting illustration of the statistical techniques involved, or be a starting point for more advanced algorithms to address image generation.

The wider context of this project is the current set of algorithms for object detection within images based on shape because this is the area where image shape has been most thoroughly considered and where the way of judging shape similarity that is used in this project will be sourced. To be more specific however, we will be working in the context of Naïve Bayes Nearest Neighbour models. These Naïve Bayes approaches are simple and tend to have good performance, as is detailed in (Boiman, Shechtman, Irani, 2008). This thesis adds a series of ways in which these existing Naive Bayes methods can be used to model the properties of a shape necessary for image generation, like the probability that two parts of a shape occur together.

To do this an existing nearest neighbour object detection algorithm has been partially implemented (Terzic, Mohammed, du Buf, 2015), however focusing on line drawings instead of complicated natural images. This limitation is in order to narrow the scope of the project and thus allow a more complete implementation. This is then extended based on theoretical reasoning to create image hallucinations. Any modifications have been tested, and we will observe how this influences the result of the algorithm. The implementation was written in Python and utilises the OpenCV library among other standard Python Libraries.

This project was completed using software that is freely available and can be used on the ordinary lab machines. The python was version 3.6 and was coded in Windows.

2. Similar Areas of Work and Related Literature

These works tackle similar problems, provide similar techniques, or illustrate a feature of the more general image generation landscape. For each, I discuss the similarity to the current problem, the limitation of its application to the current problem, and the way in which it informed my approach.

Content-aware image restoration or patch technologies such as that used in Adobe Photoshop primarily attempt to modify existing images by interpolation. This can be used to infer textures from surrounding areas onto a gap in the image or to sharpen textures on a part of an image. This is image generation of a sort, as quite large sub-images can be generated, and is probably also the best known and most common use case. While this does not extend to image generation in the context of

things like original shapes, the ways in which these content-aware technologies are evaluated is highly relevant to this project as selecting the method of evaluation is a challenge that was faced. In some circumstances, a test scenario can be generated from artificially changing an image by cutting a section out and then quantitively measuring the algorithm's ability to fill it back in in the same way that it had originally been. Alternatively, shading can be applied to a region, and similarly the ability for the algorithm to reverse this can be quantitatively tested. It is also the case that these algorithms are to some extent assessed on purely qualitative measures. Some shading techniques do just appear more subjectively pleasing for example. The first type of testing is not really possible for the problem addressed in this project, however the use of qualitative assessment is striking, as that is a method that can capture our intuitive reactions to the well-formedness of a shape, even though it cannot be automatically or precisely assessed. (Ding, Tong 2010).

In 'A Stochastic Grammar of Image' (Zhu, Mumford, 2006) the authors explore the possibility of a stochastic and context-sensitive grammar of images represented by an and-or graph. The purpose of this grammatical approach is image parsing and so is not a solution to the question of image generation. However, it does address the related question of modelling the general concept of a shape, a shape being instantiable in multiple different ways. This model of the concept of shape is one that extends to be able to explain some aspects of the image generation problem and so a grammatical approach to the very idea of shape generation is discussed in a later section. In the approach detailed in this paper, the authors first annotate image parts to be used in their grammar by hand, they do this due to the difficulty of automatically choosing what should be considered a part of an image for grammatical purposes prior to the creation of a grammar. This annotation is not a technique that is used in this project as it would not represent an extension on the Nearest Neighbour Contour Fragment approach to defining image parts. An approach that more directly creates a grammar for a shape would have to put a large amount of effort into defining and searching for shape parts suitable for use in a grammar. The idea of a grammar is used in this project only as a framework for understanding and defining image generation methods and as an explanatory tool for expanding upon the intuitions of the methods used. Additionally, these shape parts are not parts of a contour as they are in the specific problem tackled by this project (and what can thus easily be described by sampling and testing contour fragments), but proper shapes in their own right which are harder to capture using contour fragments. For other work on grammars in images see Chanda, Dellaert (2004).

(Terzic, Mohammed, du Buf 2015) 'Shape Detection with Nearest Neighbour Contour Fragments' is the paper containing the Bayesian Nearest Neighbour method which we are modifying. It is a method for shape detection, not image generation and uses the validation metrics of shape detection algorithms. The most relevant part of this shape detection algorithm and the part that will be replicated is the sampling of shape fragments and the generation of their descriptors from an image (Belongie, Malik, Puzicha, 2002), and also the use of approximate nearest neighbour to calculate the relevance of a fragment to some class. The calculation of discriminative power and the implementation of bounding boxes is more specific to shape detection and so is not implemented. An explanation of descriptor generation and the calculation of relevance is given In Section 4.1.

Full image generation is not a very well explored area, but Generative Adversarial Networks (GANs) have been turned to this purpose, with the most well-known examples being DeepDream and style transfer methods (Gatys, Ecker and Bethge, 2016). These networks, especially style transfer networks can be interpreted as advanced image interpolation that aims to take into account the more general patterns of an image through Deep Learning, as opposed to using the simpler and more limited mathematical approaches such as those used in Adobe Photoshop's Content Aware algorithm. GANs first train a deep neural network to recognise images of a certain class and then use this classifier to judge and train a second network's attempts to generate a new image, at the same

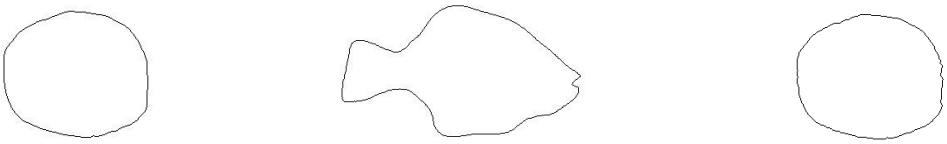
time improving the ability of the first network to differentiate by adding these new images to the training set. Once trained a GAN should be able to generate arbitrary new examples of an image class given some sort of seed input. For DeepDream this was used to generate patterns on an image for the purposes of exploring the kind of pattern that a deep neural network would usually find in an image. These DeepDream images do not look particularly coherent, much more similar to the work of this project is the later style transfer networks, where a pre-trained network supposedly takes the style of one image, including the shapes within it, and transforms a second one to have that style.

An example could be this transformation of an Isaac Asimov cover art 'robot dreams' by Ralph McQuarrie (1986), with the style of 'Transverse Lines' by Kandinsky (1923). This transformed image was created as an example for this project existing GitHub code (Cheung 2017):

Original Image | Style Image | Resultant Image

Examples like this are extremely impressive, but the question remains as to whether this approach can extend to deliberately chosen shapes. Unfair as it is, the result of running this same code on a circle in an attempt to see if using another shape as a style image might transform it into the target shape can be seen below, it does not. Apart from GANs, there is really no comparable original method for consistent content generation, as such it is not possible to compare the results of this project with the results of some competitor algorithm.

Original Image | Style Image | Resultant Image



(Attempting this transformation on an initially blank image leads to very little or no change also)

'Free-hand Sketch Synthesis with Deformable Stroke Models' (Li, Song, Hospedales, Gong 2017) comes close to sharing an objective with this project. The authors develop a generative model to create approximations of freehand sketches. This is done by studying human strokes and creating sketches through a model based on this data. The authors achieve this through perceptual grouping on the training sketches where human strokes were recorded, then matching these to parts of an edge map of a single image of the class. Finally, this edge map is deformed to create a sketch. Therefore, this is not quite the same problem approached here as the images are not a new instance of the class, but

rather the problem is that of photo to sketch stylisation. Photo to sketch stylisation is a separate problem where given an image, the image is remade in the style of a sketch of the same content. In the stylisation problem the structure, and to a large extent the shape, of the new image do not have to be generated. As such the results are not directly comparable, however their assessment methods may be able to inform the assessment methods of the problem that this project addresses as they are operating on a problem where the final result can in some degree only qualitatively be assessed in the same way.

The authors of 'Free-hand Sketch Synthesis with Deformable Stroke Models' approach its evaluation in a few ways, first they compare their results to other similar results in sketch synthesis. This is not very helpful for this project due to the fact that there are no similar results of the correct kind.

Secondly, they conduct studies with twenty or so participants to attempt to assess quantitatively the judgements the participants make of their sketches. They include a study to evaluate whether their sketches convey semantic information sufficient to be recognised as sketches by humans. The results of this project do not need to look like sketches rather than edge maps of photos, and the ability for a shape generation algorithm to have results indistinguishable from results generated from shape edges found in photos is slightly unrealistic. However, another study asks participants to classify synthesised sketches by class, in order to test the intra-category difference in their sketches, this is a more promising approach. In the paper, the test performance was highly variable with scale, and given the constraints of my project a test of a sufficient scale is not practicable, although it would certainly be a consideration if further work was done. The authors dedicate the largest amount of space in this paper to a qualitative comparison between sketch stylisation methods, and the ability of their deformable stroke model to create a realistic imitation of a simple human sketch is highly impressive. In presenting the effects of changes in hyperparameters on their algorithm the authors also use qualitative comparison instead of quantitative methods like questionnaire studies. Assessing this paper thus leads to an overall increase in the level of confidence with which the idea that qualitative comparison will be able to provide a consistent metric with which to judge the results of this paper can be presented, due to their predominant use of qualitative assessment.

3 Theoretical Foundations of Image Generation

In this section the context in which the methods of image generation that are employed sit is briefly presented. The purpose of the set of methods as a whole is explained in a more precise sense and on a wider variety of levels than simply saying that the methods make new pictures. The framework for evaluation that this project employs is also discussed. This information in this section is related to the methods and ideas in sections 4 and 5 and explains the way they were seen as fundamentally related throughout this project.

3.1 Modelling the Generation of Images

The computational representation of an image is usually a series of pixel values. Unfortunately, this representation says nothing about the conceptual content of the image as a whole in terms of the objects that might be identified by a hypothetical human interpreter, which are what we wish to re-create. Objects and their shapes do not necessarily have a uniform colour or texture, or even a relatively uniform outline, and pixel values definitely contain no explicit reference to an object as a whole. This is problematic as our generative methods occur on the level of the computational representation but aim to capture the conceptual contents.

A hypothetical representation that describes both the computational specification of an image and a formalised description of the conceptual contents in terms of the objects involved would be a context-sensitive grammar as is mentioned in (Zhu, Mumford, 2006). The natural language concepts that underlie the notion of an object and its individual identifiable parts being represented in the high-level rules of such a language, with the most basic rules being testable against a low-level pixel representation of an image.

When approaching a generative problem in this landscape we are trying to model the rules of the language and grammar that govern well-formedness of certain shapes (a shape being well-formed according to some grammar meaning that it is a good example of some class of shape). The generative grammar that represents such rules is likely to be highly complex and context-sensitive.

The reduced problem that is approached in this project is that of shape generation given examples of said shape. Where shape is a 2-d representation of an object. Although there are shapes that will have holes (e.g. a silhouette of a mug) when thought of in a 2d space this is reduced to shapes without holes for the sake of simplicity. Equivalently the problem of shape generation has been assumed to be one of single contour creation and/or selection. Where the model used for understanding the idea of what unites shapes in a class is in some form, a generative grammar across a contour line. Then the methods used in creating contours likely to be examples of that shape can be understood as capturing properties and predicting the behaviour of the generative grammar from the observed examples through the probability estimation strategies of Naïve Bayes Nearest Neighbour approaches to image recognition.

The common justifications of traditional function approximation methods are a little difficult to place when one is thinking of the problem as one of a following a grammar consisting of complex rules. This does accord with the fact that creative image generation is generally seen as a very difficult problem to even approach from the perspective of traditional Machine Learning techniques which uses these function approximation methods. I will use a function approximation method in section 4.3.3 and its lack of consistent detail largely bears these concerns out, however as a partner to other methods it still brings the comparative advantage of being able to draw a mean from all the examples to get a very generalised shape that contains information about the essence of some class of shapes.

3.2 Evaluating Generated Images

For judging the output of a generation model the fact that when judging the value of the results we are unavoidably qualitative must be confronted. The problem of creativity, especially conceptual creativity is seen as very human, due to its holistic and subjective nature. There is going to be an element of subjectivity when it comes to deciding whether one shape or another is a “better” representation of some class. Quantitive measures can be introduced through aggregating reported qualitative responses, but this does not really change the underlying fact. For the purposes of this project, the problem of image generation is taken to be at its most interesting when the qualitative nature of the problem is not ignored or dodged. As such, results will be discussed qualitatively, and adjustments to the algorithms were made primarily in order to make the images look like they *should* belong to some class. Quantitative measures like inter-discriminability can certainly still inform our assessment of our results, but they will not be primary.

Nevertheless, even though the assessment is qualitative that is no excuse for not discussing in advance the sort of criteria that will be used for assessment in more detail than saying judgements will be made on the basis of ‘does it look like x’. We have the questions of What constitutes a mistake? what determines the severity of a mistake? what is an accurate generation? what is un-

original? Do we need to be reliable? Is there a tradeoff between features that are positively desired and mistakes that make some mistakes acceptable if there are positive features included?

Generative mistakes with images will be sorted into two vague categories, one that could be seen as minor noise, affecting the degree to which a result could be said to be a good example of some shape and one which represents a major incorrect choice in generation which affects whether some generated shape could be seen as a proper example of the intended class at all. Almost any number of errors of the first kind is acceptable to avoid the second, and while there are few positive features that can redeem the second, the addition of positive features may well be worth the first kind of error.

For the purposes of this project, a tradeoff whereby some mistakes are justified so long as positive original features are created is explicitly endorsed, and this extends to allowing an approach whose results may be fairly un-reliable so long as good images are also created at a high enough rate alongside the bad. Justification for this comes in one principle way which can be sloganized as 'Generation is harder than Discrimination'. The problem of excluding bad images or image segments from counting as part of a class is not as hard as generating the good parts. If there is any heuristic to separate images that have demonstrably failed to successfully generate examples because of major mistakes or differences versus convincing new generated images exists, then an amended algorithm where some number of images are generated and one of the non-failures is outputted could easily be implemented. A consistent and fast algorithm for image generation with no small noise mistakes should very much be seen as secondary in my opinion to one that can create new positive features that pass some sort of originality test. Such an extended algorithm, alongside existing interpolation and shape recognition techniques, can then be applied to create a more reliable approach. There is ample precedent for the acceptance of this trade-off, human ability to creatively generate original content usually comes with a number of false starts and failed attempts. Ask most people to sketch a horse for example and multiple mistakes are likely to be made before anything approaching a realistic sketch is made.

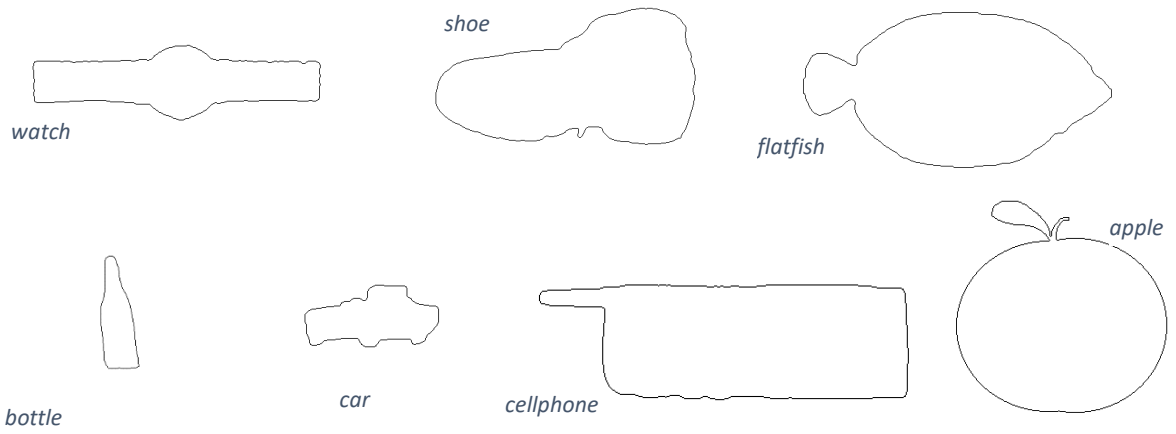
4 Design & Implementation

4.1 Re-Implementation and Adjustment of Previous Work

4.1.1 Details of Data used

This project uses the MPEG7 dataset (Lateki MPEG7). This dataset is of 2d shapes, with multiple classes of image included, such as images of various animals, everyday objects and even abstract shapes. The pictures in the dataset consist of two regions, the shape itself and the background. The shape outline is the contour between these two, it is this contour that is generated for new images. This project is mainly based on this dataset because it provides a wide variety of classes that are of the level of complexity that this project focuses on, small simple static objects almost all of which can be described by a single contour.

Here are some MPEG7 Examples:



Many of the classes have constant position and orientation, though some do not, and some classes will have a largely constant position and orientation with a few images that are exceptions to this. Many of the methods present in this project should be resistant to the existence of exceptions to a constant position and orientation as long as there is a dominant orientation and position that can be found. The classes which have no constant position and orientation could undergo some sort of pre-processing in order to create a more uniform series of shapes, but since this is auxiliary to the central problem of shape generation that this project attempts to solve, adjustments to classes where there is not dominant orientation, position or size were not prioritised.

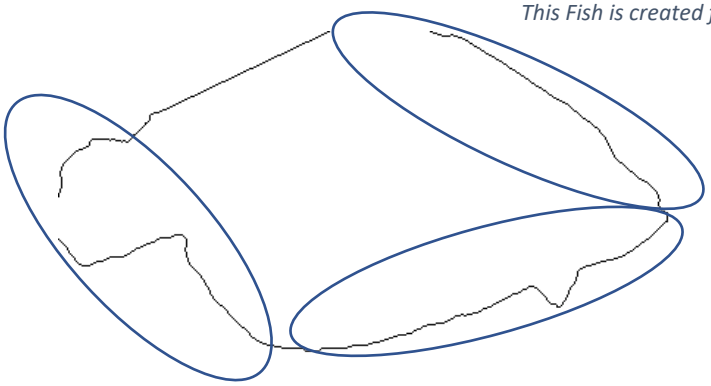
4.1.2 Fragment and Descriptor Generation

The fragments extracted are randomly chosen parts of a shape's contour. The creation of image fragments is not a neutral step in the process of creating an algorithm for image generation based on shape descriptors as it is an important sampling step. Making all possible fragments have an equal chance of being selected for inclusion in the set of fragments that are chosen to represent a shape is a possible strategy, as is including all fragments. However, by having maximum and minimum bounds for the length of a fragment (with the length of a fragment being defined as the number of points in the contour between the two ends of the fragment) the sampling range can be focussed on the type of fragment that is most useful. Excessively small fragments might not capture

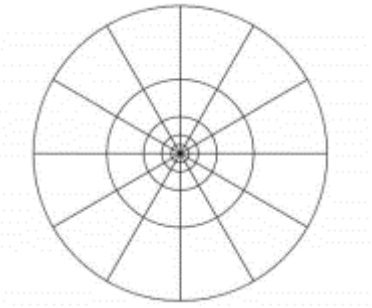
more than a small section of straight line or a corner or curve and so are unlikely to capture information as to what makes a certain shape.

Excessively large fragments will capture features that are semantically relevant to the class of shapes as a whole, but if it is also desired to use our sampled fragments as a basis for a new image, fragments that are too large will lead to new images that are obviously just a few old images stitched together.

This Fish is created from three large peices



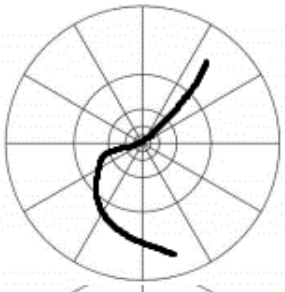
While it is ultimately a value judgement what counts as a new, original image, for the purposes of this project it was judged that making an image that was easily identifiable as being just a couple of existing images was not a suitably ambitious level of originality. If large fragments are used, as much as half of an existing shape could be directly present in the original positions in the "new" image. A method for creating such images will be presented and noted in a later section however.

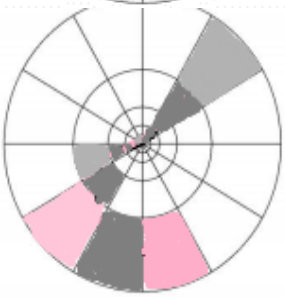


A visualisation found in Belongie, Malik, Puzicha (2002). Showing the location of the various boxes.

If the resultant images that are generated have fragments from the original images included then they would not be large enough to say that if larger relations between image parts have been successfully captured, it was simply because both image parts under consideration happened to both be included in the same large fragment. Large fragments are too close to being just the shape itself, instead of individual parts of the shape.

I created these two variation to show how a fragment leads us to a histogram:





Once fragments have been chosen, for each fragment a descriptor is generated as is described in Belongie, Malik, Puzicha (2002). This process involves choosing some number of evenly spaced points on the fragment and generating a local descriptor for each.

A local descriptor is created at some point on a fragment by transforming the fragment from x-y co-ordinates to co-ordinates in a log-polar space centred on the point chosen. The logarithmic nature of this transformation means that near points become spread out, and far points bunched.

Divide the new space into boxes and take the histogram of the number points on the fragment that fall in each box. The local descriptor is then an array formulation of the histogram.

Once all *P* histograms have been created and made into *N x M* arrays, where each histogram has *NM* boxes and has divided the log-polar space into *N* sectors and M concentric circles. The local descriptors can be concatenated to make one *PN x M* array that will serve as the complete descriptor for this fragment.

Details of Implementation

As far as the practicalities of such generation, while it does not take an excessive amount of time to extract fragments and generate them, it is still fairly significant and so in this project there is python code that generates all the fragments in one go and stores the arrays in an 'extractions' folder for later use. The number of fragments generated will have significant effects on the speed of later algorithms. Therefore, not every possible fragment between our upper and lower bounds is generated. However, if every fragment was used, many of these fragments will be extremely similar and so a more reduced set of fragments can be used with similar results. A complete extraction of fragments would then iterate through all the points in the shape's contour and extract the fragments formed by the contour sections that lead to every other point on the shape's whole contour. In the implementation used here, from every nth point, the fragment that runs along the whole contour to some subset of other points is extracted, this is made by iterating through a complete enumeration with a step greater than one, where each other point must be greater than some minimum distance and less than some maximum distance, away from the original.

The fragments and the descriptors are stored in numpy arrays within files for each class as a whole, with an additional file that contains information about how many fragments were extracted from each image and in what order, so we can associate any given contour with the specific image it came from if we know its position within this file.

4.1.3 Relevance metric

The comparison of fragments using the relevance calculation found in Terzic, Mohammed, du Buf, (2015) is essential to this project. Firstly, some fragment $\boldsymbol{m}$ is likely to be found in an image of class c when the conditional likelihood $\boldsymbol{P}(\boldsymbol{m}|c)$is greater for c than any $\boldsymbol{P}(\boldsymbol{m}|c' \neq c)$. The Bayesian approach behind the relevance calculation is that we estimate $\boldsymbol{P}(\boldsymbol{m}|c)$ by using Gaussian kernels of samples closeby in the feature space. In fact, just the nearest sample is used. Thus, the estimation of $\boldsymbol{P}(\boldsymbol{m}|c)$:

$$\boldsymbol{P}(\boldsymbol{m}|c) \approx \exp\left(-\frac{\|\boldsymbol{m} - \mathrm{NN_c}(\boldsymbol{m})\|^2}{\sigma^2}\right)$$

$\|\boldsymbol{m} - \mathrm{NN_c}(\boldsymbol{m})\|^2$ being the Euclidean distance to the nearest neighbour of $\mathbf{s}$ of class c.

The log-likelihood is proportional to this distance squared:

$$log\boldsymbol{P}(\boldsymbol{m}|c) \propto -\|\mathbf{s} - \mathrm{NN_c}(\boldsymbol{m})\|^2$$

Relevance $r(\boldsymbol{m}, \mathrm{c})$ is the probability that a similar fragment $\boldsymbol{m}'$ appears in an image $\boldsymbol{I}_c$ of class c (Terzic, Mohammed, du Buf, (2015) consider annotated sub-Images, in this project we only consider full shapes). $\boldsymbol{m}'$ and $\boldsymbol{m}$ are similar when the distance between them is lower than a threshold $\boldsymbol{T}$:

$$r(\boldsymbol{m}, \mathrm{c}) = \mathrm{P}(\|\boldsymbol{m} - \boldsymbol{m}'\| < T|c),\ \boldsymbol{m}' = \mathrm{NN_c}(\boldsymbol{m})$$

$\mathrm{P}(\|\boldsymbol{m} - \boldsymbol{m}'\| < T|c)$ is approximated as the number of images of class c in the training set, where there is any fragment closer to $\boldsymbol{m}$ than $\boldsymbol{T}$. $\boldsymbol{T}$ is estimated as the nearest neighbour of $\boldsymbol{m}$ any image not in class c. Thus,

$$r(\boldsymbol{m}, \mathrm{c}) \coloneqq \frac{\sum_{I_c} \varphi(\boldsymbol{m}, \boldsymbol{I}_c)}{N_\mathrm{c}}$$

Here, N_c is the size of c, and $\varphi(\boldsymbol{m}, \boldsymbol{I}_c)$ is:

$$\varphi(\boldsymbol{m}, \boldsymbol{I}_c) = \begin{cases} 1 & if \ \left\| \boldsymbol{m} - \mathrm{NN}_{\boldsymbol{I}_c}(\boldsymbol{m}) \right\| < T \\ 0 & otherwise \end{cases}$$

The method as implemented is this.

- For some fragment find the k nearest neighbours ordered by the distance between the descriptors.
- Find the nearest descriptor of a fragment from an image not in C.
- For the set of fragments whose descriptors are closer, find the number of unique images which have fragments present in this set, the number of such unique images is the relevance score

Consideration was also given to using a converse of this $r`(\boldsymbol{m}, \mathrm{c})$ where the first step is the same but then:

- Find the nearest descriptor of a fragment from an image in C.
- For the set of fragments whose descriptors are closer, find the number of unique images which have fragments present in this set, the number of such unique images is the new score

This second score could provide a heuristic which when a fragment is not relevant to a certain class, tells us how relevant it is to the class C', all things not in C. By picking fragments with low relevance to C' it might be possible to pick fragments that, while not being uniquely characterising of C, are at least found to some considerable degree in C.

Details of Implementation

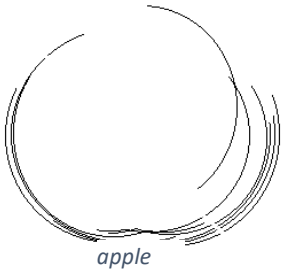
The calculation of nearest neighbours is done using FLANN (Fast Linear approximate nearest neighbours: Muja, Lowe 2012). FLANN calculation is extremely fast, and creating an index for search takes more time than running the actual search itself, this index usually only needs to be created once for a large number of searches, and so is not relevant to any asymptotic complexity of the generation algorithm. The FLANN search, although fast might still need to be run many times if there are large numbers of fragments whose relevance is tested, and so is thus a significant factor in the speed of the algorithm.

4.2 Fragment Selection

4.2.1 Selecting purely based on Relevance

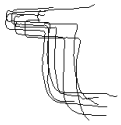
The first approach to choosing some fragments that can be combined make a shape that was tested was just to select the few most relevant fragments. While this will not create a shape that will be well-formed in the sense that it makes a complete contour, which is a property that is eventually required, it does allow us to see whether the notion of relevance is consequential enough and captures enough information about the shape to be used as the basis for a shape generation algorithm.

The results of picking the N fragments with the highest $r(\boldsymbol{s}, \mathrm{c})$, constitute a set $R(c)$, for some class c. The results $R(c)$ are presented below for three of the MPEG7 classes.



apple

Fountain



Cellphone

evaluation

These images do capture some of the more complex and unique parts of their respective classes (the fountainhead, the cellphone antenna) but miss out on large parts of the shape. When the images are already centred and have consistent orientation, while not a coherent shape, the result definitely suggests at least part of the outline of each shape. Emphasis is placed on the features that make a shape stand out from the other shapes as is expected due to the use of relevance.

This result is encouraging in the sense that when searching for relevant sections of a shape the fragments found clearly were what a relevant section ought to be if the theoretical justification for the relevance metric is correct, i.e. the most complex fragments that are consistently necessary to produce a shape. However, a problem also arises in that when choosing simply based on relevance the fragments chosen tend to be extremely repetitive. This is to be expected since if there are N fragments from class C that are more similar to some fragment $\mathbf{s}$ also from class C than from any fragment not in C, it is intuitively likely that some of those N fragments will also be more similar to at least some other of those N fragments than other fragments not in C. Thus, if this relevance score is relied on too much when deciding which fragments are “good” to use, then while including possibly the most important detail of a shape, a large amount of other detail will be missed.

4.2.2 Selecting based on Relevance and Difference

In response to the way in which a purely relevance-based approach tends to pick many very similar fragments, a new method was created where a much larger number of relevant fragments are taken, and the this set of fragments is reduced using an efficient measure of difference to create a more diverse set of fragments that can be used for shape construction.

To find a difference subset $R^f(c)$ of the selection from the previous section $R(c)$.

$$R^f(c) = \left\{ \boldsymbol{m} \mid \boldsymbol{m} \in R(c) \; and \; \left\| \boldsymbol{m} - \mathrm{NN}_{R(c)}(\boldsymbol{m}) \right\|^2 > \left\| \boldsymbol{m^K} - \mathrm{NN}_{R(c)}(\boldsymbol{m^K}) \right\|^2 \right\}$$

and $\boldsymbol{m^K}$ is such that $\left|R^f(c)\right| = N$ `, for a chosen , $N` < |R(c)|$.

Although this strategy is unlikely to find the maximally spaced subset of the relevant fragments it should return one fairly well-spaced, and due to the predominance of similar fragments in the pure relevance approach, It is unlikely that miss those previously significant but repeated fragments are missed.

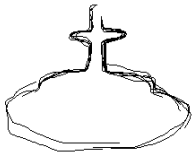
Details of Implementation

This is implemented by creating a new FLANN index consisting of the enlarged set of relevant fragments, then take some number of fragments from this relevant set that have the largest distance to their nearest neighbour as expected.

Results and Evaluation

As is clear in these examples, even with many fewer fragments, a much larger amount of the original shape outline can be present in these collections of fragments.

apple



Fountain

Cellphone

This approach is unsound as a method to achieve complete shapes due to the lack of well-formedness guarantees, what this method does provide though is a good sample of fragments from which original shapes could be constructed. The sets of fragments used in the following approaches are sets constructed in this way.

4.3 Shape Creation Methods

4.3.1 Fragment Replacement Iteration method

Having acquired a set of fragments that captures some of the features of a shape the next challenge is creating a shape that has the well-formedness conditions of being a complete non-overlapping contour. A simple way of ensuring that this condition is met is by starting with a shape contour that already meets the well-formedness condition and then iteratively changing it towards the chosen goal, whilst at each step preserving the well-formedness condition.

This is pseudocode for such a repeated replacement:

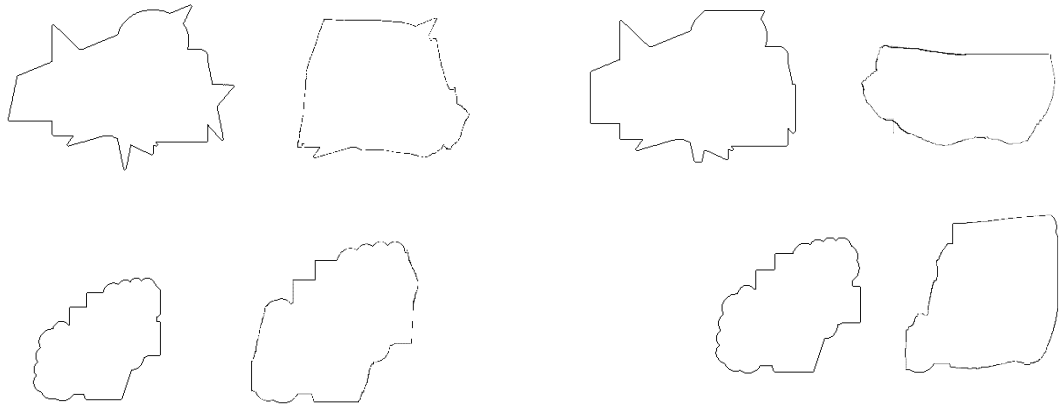
```
Fragment Replacement (Takes a shape s that consists of a list of points):
        Point a = some random point in s
        While True
                Point b = some random point in s
                Distance = smallest number of points in between a and b (clockwise or
                anticlockwise)
                If chosen min < distance < chosen max
                        break
        list of candidates = select n random fragments from a list of fragments from class c
        list of candidate shapes = new empty list
        for each candidate fragment
                angle = angle from a to b
                rotate candidate so angle from start of candidate to end == angle
                size = distance from a to b
                resize candidate so size of distance from start of candidate to end == size
                final candidate = resized candidate moved to fit between a and b
                candidate shape = all points a to b in s replaced by final candidate
                candidate shape = fill missing points and remove duplicates(candidate shape)
                append candidate shape to list of candidate shapes
        for each candidate shape in list of candidate shapes
                randomly select n fragments of candidate shape where the fragments include
                candidate fragment that was inserted to generate this shape
                total relevance = 0
                for each fragment selected
                        total relevance += relevance(selected fragment)
        let the new shape be candidate shape with greatest total relevance
        repeat some chosen number of times
```

The motivation behind this approach is that by testing a wide variety of fragments not only including the replaced section but also including the surrounding area as well, not only is the chance of the new fragment being a part of any given shape of this class measured, but the probability that it accords to the grammar that describes the class and thus stands in the correct relation to the parts

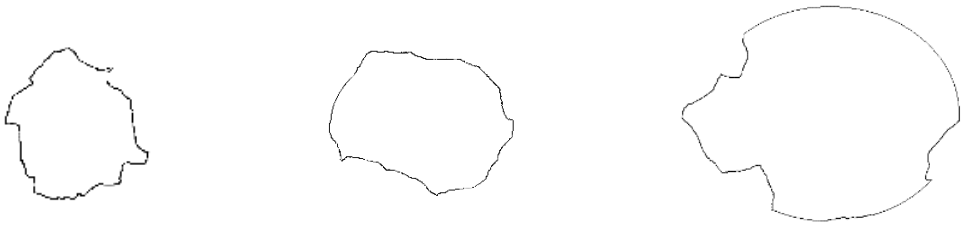
around it can be estimated. This happens by taking fragments which overlap the new fragment or include the whole of the new fragment along with other parts of the image.

Results

All these images were generated through the iterative replacement approach, the first four have the original shape on the left, and the transformed on the right, the other three were generated from a simple circle. The target class was the flatfish as can be seen in the examples above or section 5.



These were generated from circles:



Evaluation

From a qualitative assessment of these examples It was possible to identify several trends in the results that tend to be generated by this method:

- The final shape that results is highly determined by the shape that was begun with, as can be seen In the circle examples remaining roughly circular. There was no indication that it was possible to transform the larger features of one shape into the larger features of some desired shape using this method. The very last shape appears to have developed a sort of fishtail but the addition of correct features was not sustained through additional iteration.
- The shapes tend to decrease in size as replacement occurs, this can be caused by a set of fragments used in replacement when those fragments tend to be fairly straight, as repeated replacement will cut chords across any general arcs of a shape.
- Features of the original shape that are completely incongruous with the class targeted are often likely to be replaced at a faster rate than those that are consistent with some part of the desired shape. This is the most promising feature of the method and suggests that if a method was created that was able to introduce these large features, even if it also creates implausible features as well, it might be able to use the shape created through that hypothetical other method as the basis for this style of replacement and generate generally plausible shapes.

These trends suggest that information on the rules governing the creation of shapes of some class, available as the observed relations between parts of a shape that we hoped to invoke by sampling fragments that connected replaced fragments with the surrounding image parts and then testing relevance, is not invoked using merely this replacement method in a significant enough way to allow this method to stand alone as a way of generating plausible shapes of some class.

It is not the case that this necessarily implies that such information is not significant at all, as it could be rather that the search frontier is not big enough. A search frontier that is too small and cannot reach the desired shapes from the starting points given by the unrelated shapes is a better explanation. Since but a few alternative fragments are considered, and it is required that each fragment is instantly an improvement, and only one image is kept at a time the search frontier is small and the ability to jump within the search space is limited. It is highly unlikely that with any method for evaluating how good any of these replacements, it would be possible to achieve a convincing shape of the desired class.

Unfortunately, the limitations of the search method are largely imposed not due to lack of imagination but due to efficiency constraints upon the implementation itself. These arise in a few places: Due to the time consumed in transforming each point in the fragment so it will fit in the location, orientation and size required, as well as filling-in any gaps created due to the change in size, as well as checking for points that were transformed to the same point, due to the necessity of checking that a replacement does not create a crossover and so violate well-formedness (for these fragments that requires checking that no two points are the same in the new fragment and the remainder of the old shape), due to the time taken in sampling fragments from the new shape, creating their descriptors, and then testing each against a FLANN index, and calculating the relevance score for each.

4.3.2 Very greedy addition method

Another method which takes the alternative approach to well-formedness constraints is one that connects existing fragments. A preferred fragment is picked, then a shape created by adding another relevant fragment to an image containing the first, then joining the two up to create a full contour. Next, more relevant fragments are added into the image, changing the way they are joined up to keep a full, non-overlapping contour at each moment, till we have a result that is of sufficient complexity or no new fragments to add to our shape while ensuring our well-formedness requirements obtain can be found. The goal being to end up with good ways of creating images by joining-up some compatible subset of chosen relevant fragments. Ultimately there will be many different ways of exploring the search space created by this approach, but initially, a simple version of this method will be used:

- Create a set $S = R^f(c)$ as in 4.5
- Select the fragment $\boldsymbol{m} \in S$ where for all $\boldsymbol{m}` \in S \; and \; \boldsymbol{m}` \neq \boldsymbol{m}$

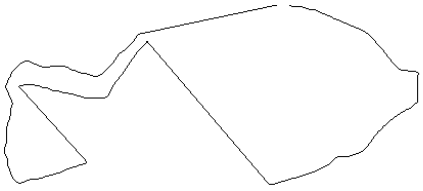
$$\|\boldsymbol{m} - \mathrm{NN}_S(\boldsymbol{m})\|^2 > \|\mathbf{m}` - \mathrm{NN}_S(\boldsymbol{m}`)\|^2$$

  Remove $\boldsymbol{m}$ from S and add it to a new image $\boldsymbol{I_{new}}$
- Repeat this selection method to select each fragment in S till S is empty, each new fragment $\boldsymbol{m}$ is added to $\boldsymbol{I_{new}}$ in the following way.
- If $\boldsymbol{I_{new}}$ contains one fragment $\boldsymbol{m^1}$ and there is a new fragment, $\boldsymbol{m}$ then add connecting lines from one endpoint on $\boldsymbol{m^1}$ to one endpoint of $\boldsymbol{m}$ and the other remaining endpoints are also connected.

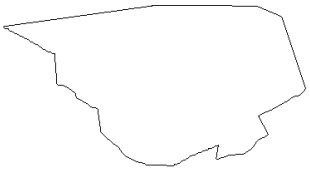
- If there is no way of achieving this where these new lines do not intersect with $\boldsymbol{m}$, $\boldsymbol{m^1}$or each other, or $\boldsymbol{m}$ crosses $\boldsymbol{m^1}$ do not add $\boldsymbol{m}$.
- If $\boldsymbol{I_{new}}$ does not just contain one fragment then it will contain a series of fragments and connecting lines $\boldsymbol{I_{new}} = \boldsymbol{m^1 c^1 m^2 c^2 m^3 c^3 \ldots m^{n-1} c^{n-1} m^n c^n (m^1)}$, where $\boldsymbol{c^i}$ is the line connecting fragments i and i+1, the order of fragments here is not necessarily the order in which they were added. $\boldsymbol{m^1}$ is added at the end as this is a contour where the final connecting line must connect to the first fragment. To add a new fragment we will remove some connecting line, the result is no longer a contour but can now be thought of as a single fragment $\boldsymbol{m^*}$, we can now add our new fragment $\boldsymbol{m}$ to $\boldsymbol{m^*}$ in exactly the same way as we did in the case where there was only one fragment in $\boldsymbol{I_{new}}$, with $\boldsymbol{m^* = m^1}$.
- At this stage, we will add each new fragment in the first way we can find which causes no intersections

Results and Evaluation

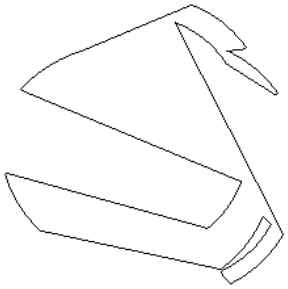
This method can create fairly complex contours of a desired shape. The hope is that if the shapes are relatively uniformly positioned one will be able to create a shape that is a convincing example of the class by simply joining some grammatically correct subset of fragments found in 4.5. However, as a greedy search method, it is unlikely to find an optimal solution to the problem of adding contour fragments together. Here are some of the less inspiring examples where even though we have added a large number of fragments the resultant image is not a good example of the class:



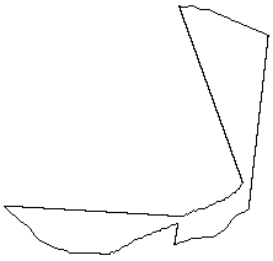
flatfish



flatfish



apple



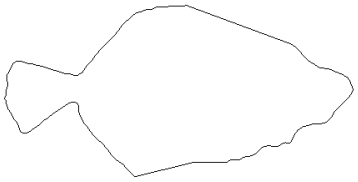
flatfish

The results for fragments of a small enough size to allow some degree of originality for images are fairly random in this greedy approach, clearly there are better ways of exploring the search space than this, but they will require a set of heuristics with which to prioritise the search. In fact, the variety of the images generated and the comparative speed when compared to method 4.6 imply that given a good set of heuristics with which to inform a search, the possibility of creating interesting original shapes is very promising, as we can quickly generate a large number of new shapes, with the possibility of also somewhat being able to capture more of the large features

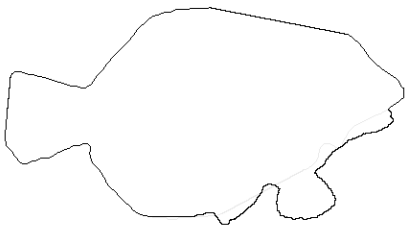
characteristic of the example classes than was possible in 4.7. I return to this method as the basis for a combined approach in section 5.

Alternative Implementation for Large Fragments

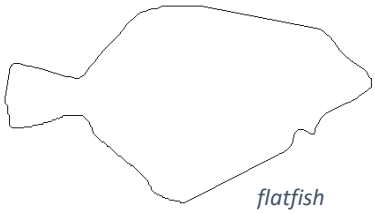
I will note however that if there is a willingness to have large fragment sizes technically original images of some class that are very faithful recreations of the shape can be created, although they are not necessarily satisfying for a strong requirement of originality. These images are the result of connecting three large fragments chosen by 4.5 and represent a valid solution – even if a relatively trivial one and not rising to the challenge of originality that is sought in this project. However, originality in the sense that each image looks like an organic and possibly slightly noisy variation on the existing examples may not always be appropriate. The problem of creating realistic original shapes of highly regular shapes like vending machines, planes and phones could best be served by a smart recombination of existing features. For the purposes of this project, these images should be regarded as an interesting, if problematic interim result. The status of these results as an interesting and complete contribution is reinforced by the fact that the creation of these images was significantly guided by the selection of fragments chosen through our relevance and difference measures from earlier sections.



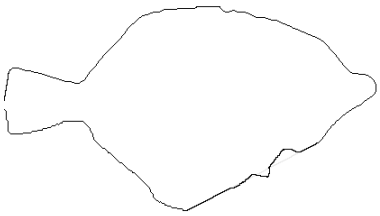
flatfish



Flatfish – four fragments



flatfish



flatfish

4.3.3 Shape Splines

A third approach is that of a Spline through some sample points from the shape, this represents a more traditional function approximation approach to the problem of image generation.

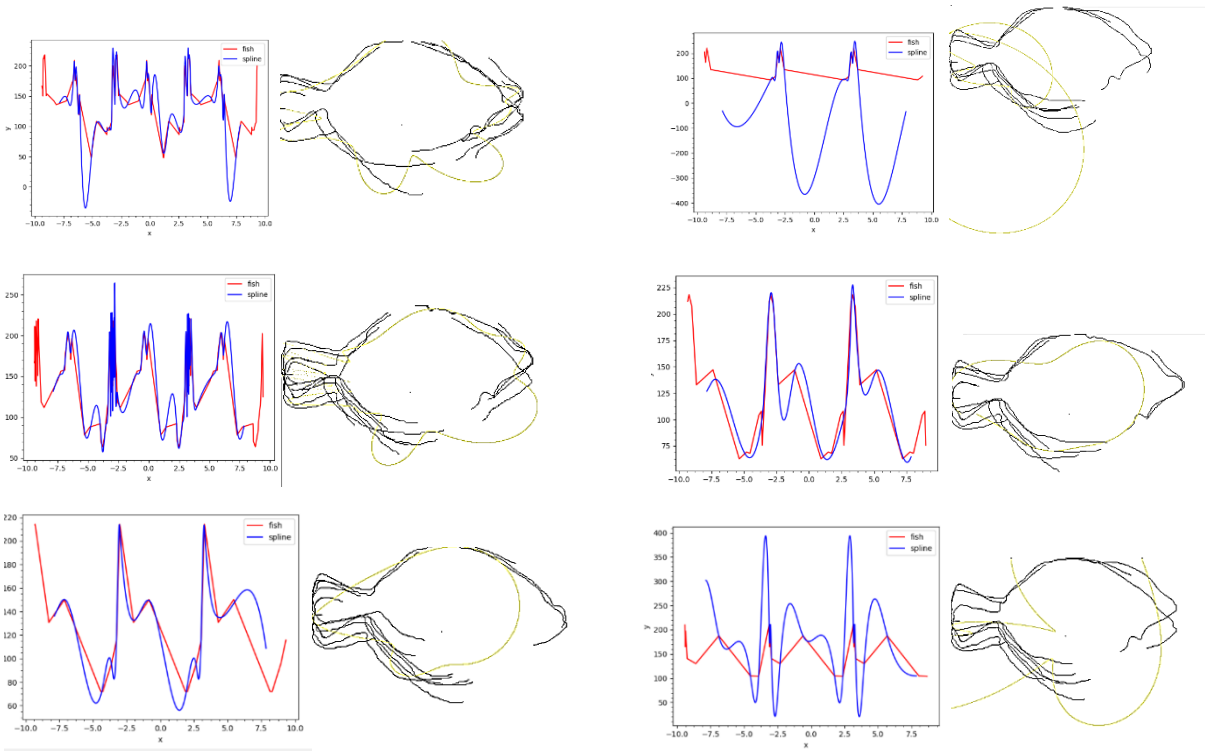
To create a spline, The mean location of all the points in my example shapes is calculted to try to estimate the centre of the shape for uniformly located shapes. Then, a sample of points from a fragment set as in 4.5 are taken and their locations relative to the central point is calculated. After this these points were converted into polar coordinates from the central point. The data was repeated twice, but with extra rotation around the centre, so that a spline interpolation would be almost always to repeat its approximation of the relevant data, and the points which represent the furthest locations away from one another in terms of the radian measure in our original sampling

would have the same distance away from the centre in the spline, as they are in fact the same point on the contour. The spline was then fit and converted back into cartesian coordinates and presented along with a graphical representation of the spline. The spline was UnivariateSpline taken from scipy.interploate.

This is the pseudocode for the generation of samples.

```
Generate Samples for Spline (for some class c):
        Total x, total y, point total = 0
        For each image of class c
                Point total ++
                For each point in contour of image
                        Total x += point.x
                        Total y += point.y
        Average point = (Total x / point total, total y / point total)
        Sampled points = every nth point from every fragment in R^f(c)
        Adjusted sample points = to_polar_coordinates(sampled points, origin = average point)
        Extra samples below =  minus 2 pi to each angle (adjusted sample points)
        Extra samples above =  add 2 pi to each angle (adjusted sample points)
        Return Extra samples below + Adjusted sample points + Extra samples above
```

Interim Results and Evaluation



From looking at these results we can see that there are significant issues with this simple spline approach.

Firstly, For fitting a spline to points sampled from fragments of a shape the qualitative nature of the end result means that choosing the desired level of smoothing versus accuracy, is not easy, and

although partial normalisation can help create a more uniform result, there was no easily accessible solution to the problem of setting the degree of normalisation and the degree of smoothness that worked for every class of shape.

Additionally, any sort of smoothing is likely to be problematic insofar as certain essential details of the object can only be realistically captured through spline parameters which would in normal circumstances be thought of as creating a near-certain likelihood of overfitting.

Also, the points sampled from fragments from 4.5 (and even if the original shapes were sampled) are not guaranteed to be uniform, leading to the existence of large gaps in the data used, allowing the method to have periods of unconstrained interpolation as error is only calculated by the degree to which the spline misses any given point. This is especially problematic since an approximation at all parts of the shape is required and so when there are large variations away from what could in any way be called a reasonable approximation of a shape the spline approach becomes a bad solution.

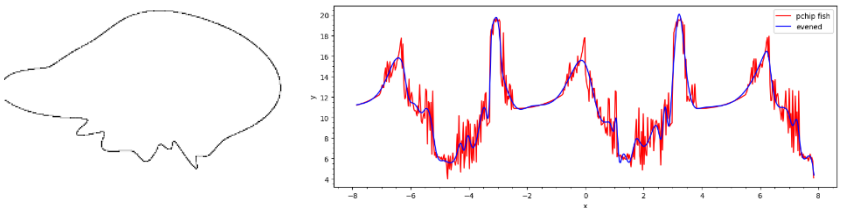
Adjusted Method

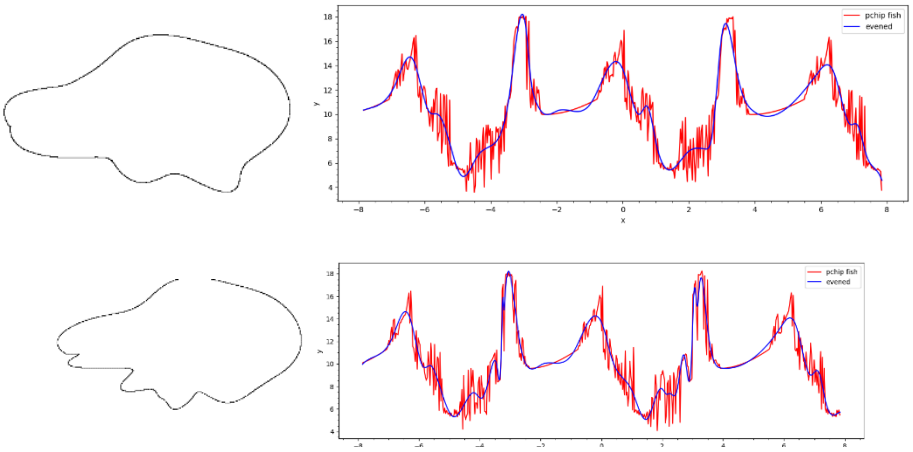
A spline method that mostly deals with these problems is just a little more involved. The method is the same as it was in the first spline approach up to the point where the spline is actually fit, however all the polar distances are first multiplied by some parameter between zero and one, in order that the spline fit is more uniform as discussed earlier. At that point at which the Spline would be fitted, instead a Piecewise Cubic Hermite Interpolating Polynomial PCHIP (Kahaner, Moler, Nash 1988) technique for interpolation is used. This new technique is used to create an initial spline that goes directly through each point of the data, as opposed to just near most of the points as the univariate spline did. From this pchip spline, a new set of points are sampled in order to fill out any large gaps between the original sampled points

```
Generate Adjusted Spline (for some class c, some multiplication parameter p)
        Initial samples = Generate Samples for Spline (c):
        Used samples = multiply all distances in sample (initial samples, p)
        PCHIP = fit PCHIP (Used samples)
        New sample angles = range(-2pi, pi, step = 4pi / desired number of samples)
        New sample distances = PCHIP(new sample angles)
        Final Spline = fit UnivariateSpline ([New sample angles, New sample Distances])
        Needed angles = range(-pi, pi, step = precision desired in spline)
        Needed distances = Final Spline (Needed angles)
        Shape Points = to_cartesian_coordinates(sampled points)
        final shape = fill missing points and remove duplicates(shape points)
        return final shape
```

Further Results and Evaluation

Here are some results from the new refined method for the flatfish shape:



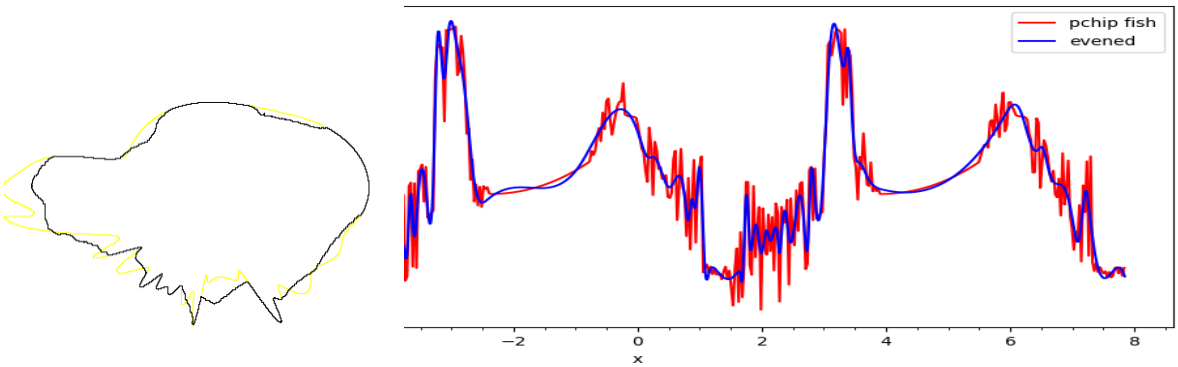


These images clearly show a more consistent outline, and they also consistently display the general nose forward shape of the flatfish class. There is a tradeoff between detail in some locations and a jagged-edge caused by overfitting in others. The first image has detail around the lower fin, and a more pointed face as should be expected, but has overfitting further to the back, whilst the second has no overfitting but also has lost these details. The third may present a sort of compromise, but to achieve this some fairly extensive qualitative fine-tuning of the normalisation parameter is required.

There is also the difficult issue of shape parts where the contour should curve back on itself from the perspective of the centre, which makes it difficult to capture the way the tail fin radiates out from its base. This is more acute in other shapes where this property is more significant, e.g. any sort of spiral or T-shape.

Adding a previous method

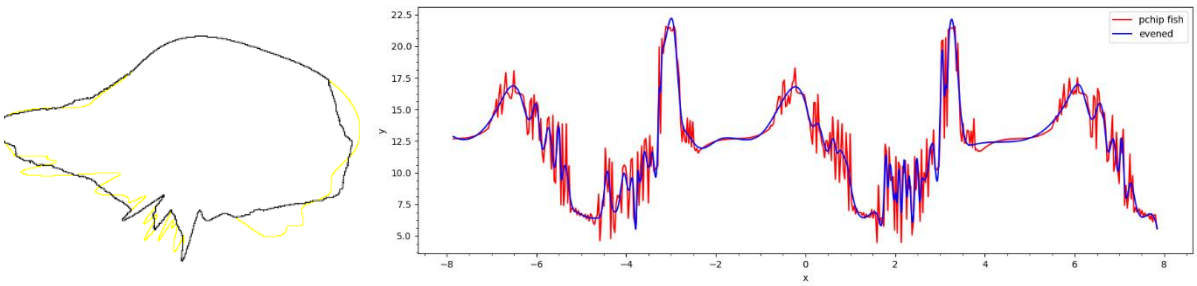
Before moving on to a combined approach where the spline method is used to aid other methods, it is perfectly possible that the spline method could itself be the basis for improvement by other methods, most notably by 4.3.1. Below is a selection of flatfish again, generated as previously, but with a few iterations of 4.3.1 applied afterwards.



With these images, significant amounts of the jagged-edge effect was allowed in order to create more detail and interesting features. This jagged-ness has been much reduced in the iterative replacement stage, although significant amounts still remain. Features like the distinctness of the

tail-section, the pointedness of the nose have also been reinforced by the replacement step which is extremely pleasing.

In this second image, the replacement was more aggressively applied, and the spline was slightly more heavily normalised. Here we can see that some of the detail and shape that was present in the first image has been lost as a result. This suggests that the ability for the spline approach combined with an iterative replacement approach at the very least requires significant qualitative fine tuning also.



4.4 A Combined approach

Having tried addition, replacement and spline methods for creating shapes, and having discovered the limitations and possibilities of each, These will now be combined in a single method which is used to create original shapes.

The central construction strategy for the shape is a more advanced version of the method in 4.3.2, using small fragments to allow for originality and a more sophisticated exploration of the search space. Since this approach to generation results in fragments being connected by unrealistic straight lines, the replacement method of 4.3.1 is used to choose fragments that are more realistic to replace these straight lines with once the shape has been created. The spline approach in 4.3.3 is used along with other heuristics to guide the exploration of the search space.

Every time a shape $\mathbf{s}$ is generated, a spline approximation of the shape is generated first. Then, when assessing a shape to determine its priority in our search, we calculate the total distance of each point in our new shape to the nearest point in the spline (and vice versa) to create an estimate of the distance between the two $D_P(\mathbf{s})$.

Calculate $D_P(\mathbf{s})$:

```
Int TotalDistance = 0
For each point p in shapeCreated:
        TotalDistance += ‖p − NN_splineShape(p)‖
For each point p in splineShape:
        TotalDistance += ‖p − NN_shapeCreated(p)‖
Return TotalDistance
```

if shapes that have a low total of nearest distances are created then we are likely to have a shape similar to the spline, and thus, a shape that follows the general patterns of the shape that we identified in the spline.

In addition to this similarity to the spline, also prioritised are the search for shapes that were the result of the addition of more different fragments. This is due to the fact that each fragment will likely contain only a limited number of important features of the object, and by having more fragments it is likely that more of the desired feature of the shape are introduced. $E(\mathbf{s})$ is the number of fragments used in the shape $\mathbf{s}$.

Also, the average area inside the original contours found in the example shapes of the desired class is calculated, and by minimising the difference squared between that mean and the created shape we can prioritise shapes that are of the correct overall size, hopefully preventing areas where large sections of the shape are missing, or there are superfluous areas added. These areas are calculated using OpenCV `contourArea`, the area is computed using the Green formula.

For some shape $\mathbf{s}$,

$$D_A(\mathbf{s}) = \left(Area(\mathbf{s}) - \frac{\sum Area(contour(\boldsymbol{I}_c))}{\boldsymbol{N}_c}\right)^2$$

where $contour(\boldsymbol{I}_c)$ is the shape extracted from some image of class c by taking the largest contour, and $\boldsymbol{N}_c$ is the number of images in class c

Finally, the average length of the contours found in the example shapes of the desired class is also used. Minimising the difference squared between this mean and the created shape sections might also allow us to prevent shapes with edges that are significantly more complex than in the examples even if the shape has roughly the same area.

For some shape $\mathbf{s}$,

$$D_L(\mathbf{s}) = \left(|\mathbf{s}| - \frac{\sum|(contour(\boldsymbol{I}_c))|}{\boldsymbol{N}_c}\right)^2$$

The steps to generate a new image of class c for this combined method are as follows:

- Create a set $R^f(c)$ as in 4.2.2
- A search frontier f is a set consisting of generated shapes that we have created, initially $f = \emptyset$
- Add shapes consisting of two connected fragments from $R^f(c)$ to f until there are no more two fragments in $R^f(c)$ that can be connected to make a shape, or some size limit is met.
- Create a set v to improve, v consists of the shapes $\boldsymbol{s} \in f$ with the best $D_A(\mathbf{s})$, $D_L(\mathbf{s})$ and $E(\mathbf{s})$
- For each shape $\boldsymbol{s}^* \in v$ find a fragment $\boldsymbol{m} \in R^f(c)$ such that $\boldsymbol{m}$ has not been previously added to $\boldsymbol{s}^*$. Let $(\boldsymbol{s}^* + \boldsymbol{m})$ be the result of adding $\boldsymbol{m}$ to $\boldsymbol{s}^*$ by replacing one of the current lines that connects the fragments in $\boldsymbol{s}^*$, with $\boldsymbol{m}$ and new connecting lines such that the shape constitutes a single non-intersecting contour as in 4.7. To increase the chances that this is an improvement it must be the case that either

  $D_A(\boldsymbol{s}^* + \boldsymbol{m}) < D_A(\mathbf{s})$ and $D_L(\boldsymbol{s}^* + \boldsymbol{m}) \times 0.9 < D_L(\mathbf{s})$, or

  $D_A(\boldsymbol{s}^* + \boldsymbol{m}) \times 0.9 < D_A(\mathbf{s})$ and $D_L(\boldsymbol{s}^* + \boldsymbol{m}) < D_L(\mathbf{s})$

  Add each $(\boldsymbol{s}^* + \boldsymbol{m})$ to f.
- If the search frontier reaches some size threshold heuristically select some of the least promising shapes and remove them from the frontier

- If $|f| > l$, where l is some size limit chosen for performance reasons then we choose some shapes to remove from f. Choose the 3 shapes with the greatest $D_L(\mathbf{s})$, 2 shapes with the greatest $D_A(\mathbf{s})$ and the shape with the lowest $E(\mathbf{s})$.
- Keep updating the search frontier until the addition step fails to add any new shapes

The choice of the best image from the final frontier is not something for which there is an overwhelmingly obvious approach. At this point, there are a set of effective heuristics to apply, but there is no policy regarding their relative importance or the extent to which they provide a more than just a suggested ordering on created shapes. As such we cannot at this time have the selection of our final shape from this frontier be as systematic as we might prefer. The approach that as chosen is one that emphasises the lessons learned from all the methods that have been tried up to this point and has been significantly based on experience gained through testing several different approaches:

> Given a search frontier f consisting of N shapes, where each shape $\boldsymbol{s}$ is so far associated with a comparison distance to a spline $D_P(\mathbf{s})$, a squared difference in area to the average $D_A(\mathbf{s})$, a squared difference in length to the average $D_L(\mathbf{s})$, and a number of fragments used $E(\mathbf{s})$.
>
> Create a reduced frontier f` where $f` = \{ \boldsymbol{s} \mid D_A(\mathbf{s}) < D_A(\boldsymbol{s^K}) \}$ and $\boldsymbol{s^K}$ is such that $|f`| = N`$, for a chosen , $N` < N$.
>
> Create a reduced frontier $f``$ where $f`` = \{ \boldsymbol{s} \mid E(\mathbf{s}) < E(\boldsymbol{s^K}) \}$ and $\boldsymbol{s^K}$ is such that $|f``| = N``$, for a chosen , $N`` < N`$.
>
> The chosen shape $\boldsymbol{s}^*$ is $max(f``)$ as ordered by $D_P(\mathbf{s})$.
>
> The squared difference in length to the average $D_L(\mathbf{s})$ is not used at this stage as generally I found a significant reduction was required at each step for each heuristic to remove unreasonable results and the search frontier was not of large enough size to sustain too many such removals.

This method created some results that had an impressive degree of originality and accuracy. Although due to the variety of possible heuristics, there was a formidable array of possible ways to guide a search through this space, of which only a limited amount could be explored.

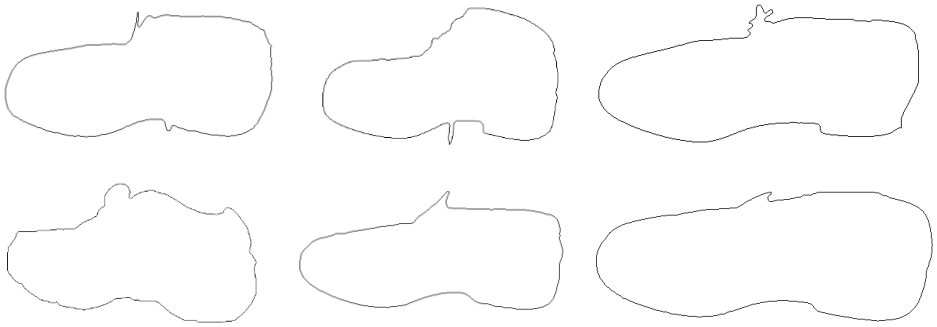
In general, limiting the size of the search frontier to somewhere between 30-50 and 1000 represented the range of reasonable trade-offs between exhaustiveness and speed that one might like to make, although even with a small frontier this search was not fast.

5 Results and Evaluation of the combined approach

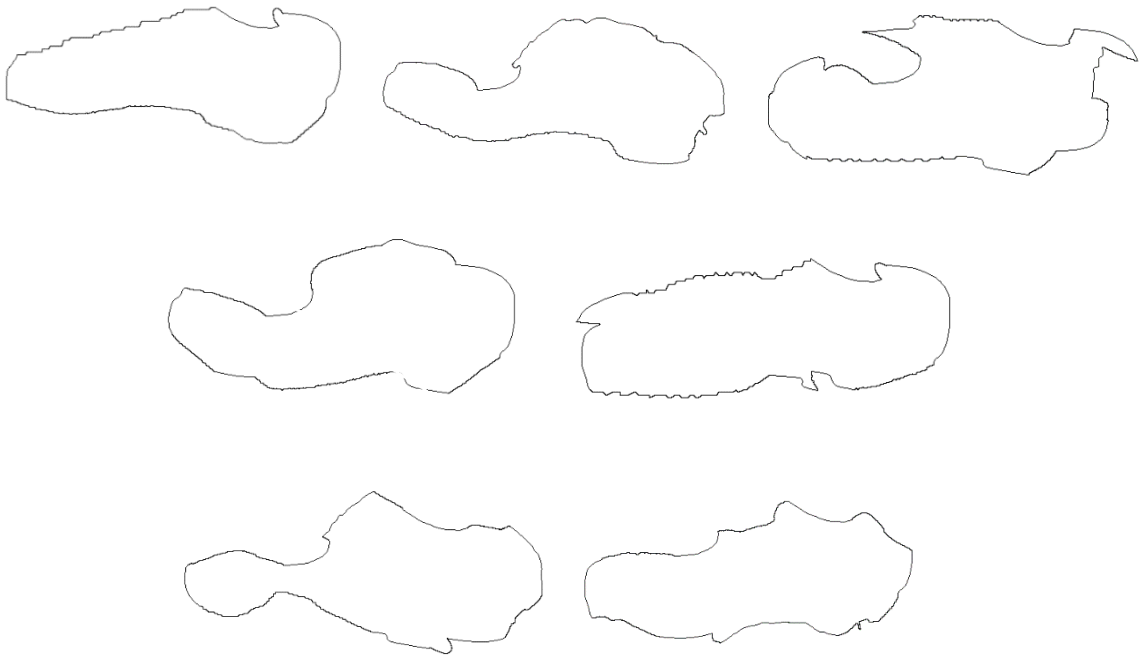
This selection consists of a larger selection of results for multiple classes. Keys, Shoes and Flatfish, along with example images from the MPEG 7 Dataset.

Shoes

MPEG 7 Examples:

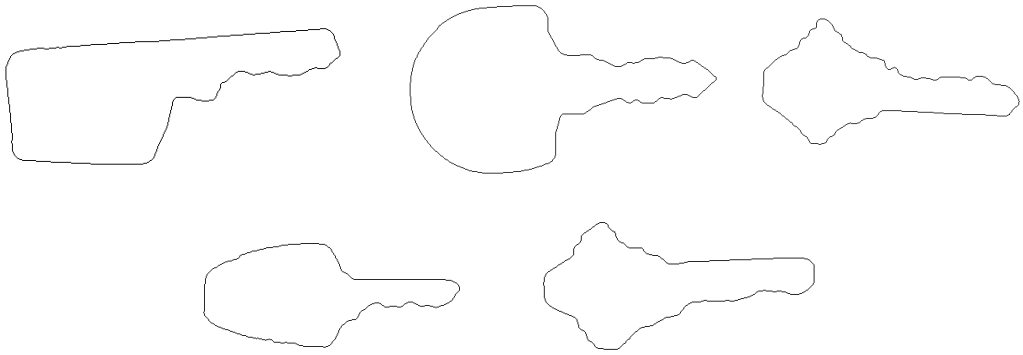


Generated Examples



Keys

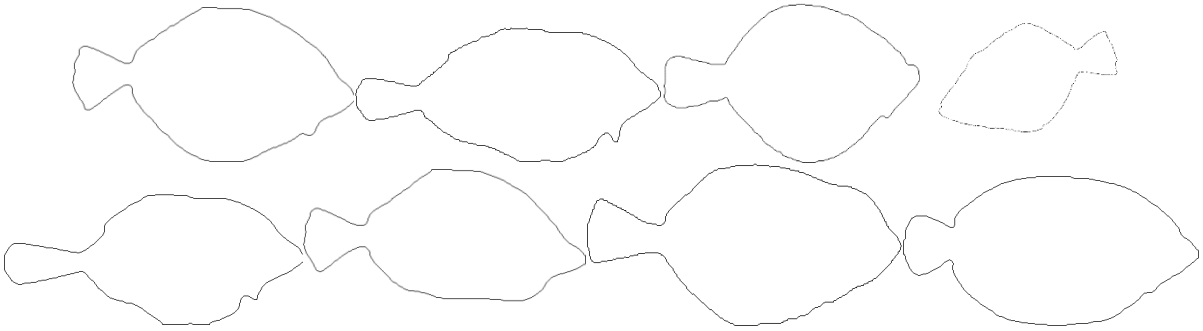
MPEG 7 Examples:



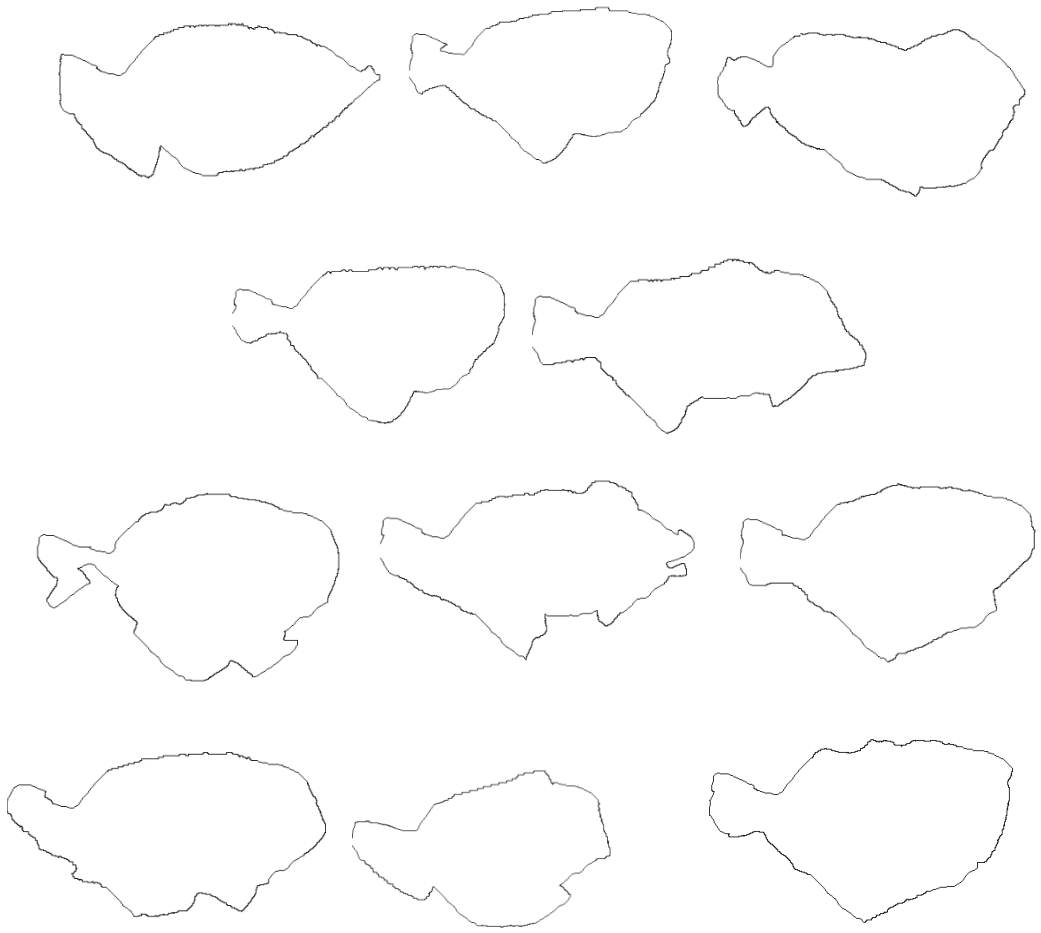
Generated Examples:

Flatfish

MPEG 7 Examples:



Generated Examples:



Evaluation

Overall the generated images are pleasing. Some do have fairly significant mistakes, and there is a general pattern to the way the shapes diverge from the example shapes that for example makes the generated shoes seem exaggerated and the keys overly complex. This represents both a general noisiness of the images, and possibly a tendency to favour busy contours in some of the heuristics I use, for example $E(\mathbf{s})$ may cause too many features to be included.

Nevertheless, there is an ability to capture the features of a shape that goes beyond any of the previous methods. In the context of generation from shorter fragments, it is clearly superior to the method in 4.3.2. Additionally, the general shape is obeyed unlike in the 4.3.1 but unlike 4.3.3 specific detail can still be captured. To some extent, these assertions are left to the viewer of these images to assess. For these three classes inter-discrimativeness is present, but there is no way to draw a conclusion as to the result if the number of categories were increased by a large amount.

These results certainly represent a kind of originality that interprets as inherently plausible a fairly large array of things which we might consider to be close to the class that is being imitated, this can manifest itself in a way which can be positive, creating fantastical variations upon existing categories as is the case with some of the shoe or key pictures. It is the nature of shape generation that also leads this same sort of variation to look like unrealistic noise in some of the other images.

6 Further Directions and Conclusion

Some further directions that could make interesting use of this project will be discussed first, along with ways it could be interestingly extended beyond what I implemented. As was noted earlier a method to estimate relative location of fragments in order to combat orientation differences would be helpful for introducing these methods to a wider range of possible applications. For example, some prediction of the expected location of a fragment relative to the other fragments already in the image, perhaps even a simple mean, would allow the creation of an initial set of fragments that would have some immunity to example classes that did not have uniform location and orientation. However, this would not have added any power to the existing methods when applied to uniform classes.

Also, I feel that additional experimentation on the search space created in section 4.4 could yield a massive variety of interesting approaches and is something that I would focus on given the ability to continue to develop this project.

returning to the relations to existing work after considering the work that this project presents of shape generation, it is more clear how object recognition approaches from a wide variety of places could be included in a generation approach. Shape skeletons for the example class could be created, and then fit to the generated images (see Liu, Wu, Hsu, Peterson, Xu 2012). Or the concept of an image grammar could be more literally applied as in Zhu, Mumford (2006) with deviation from the grammar being used as a metric to determine how different a generated image is from its nearest example and ensure that an unrealistic combination of realistic individual features is more unlikely. Both of these would be an interesting addition to the search in section 4.4.

To recap the contributions of this project, the relevant parts of an existing object detection algorithm with line images, in this case shapes, were implemented. Models which extends the Naïve Bayes Nearest Neighbour algorithm from Terzic, Mohammed, du Buf (2015) to capture and test compatibility between neighbouring segments were created and several methods that arose from this were created, implemented and tested. A combined method to generate original images was also created, implemented and tested. In addition to the analysis of this approach, I considered the differences between this algorithm and other pre-existing similar approaches to similar questions, and the potential for their combination or comparison.

The final conclusion of this project is that the Naïve Bayes Nearest Neighbour algorithm from Terzic, Mohammed, du Buf (2015) can be extended through the notions of difference, through the methods of shape generation from a fragment set born from these things, through the use of relevance to judge these generations, and the use of other search methods and heuristics to further guide this process. There are also some practical lessons learned regarding the problem of image generation which are worth remarking upon. First, that the problem of image generation is generally speaking, as hard as it is made to be. Definitions of originality and what the subject of the generation is, all matter hugely. Even in narrowed areas such as that of shapes complete reasoning as to what the nature of the real-world thing being represented in a shape plays a large part in what is and isn't an acceptable variation in that shape. Additionaly, Large Variability in difficulty occurs even in the domain of fairly simple shapes, some types of shapes can be basically immune to generation techniques that work well for others. The search space can be practically unlimited, and a crucial step in any approach similar to the one I took is severely limiting the search in some way that retains accuracy. This was automatically done in this project in the form of the fragment selections in 4.2.2, but this selection still proved insufficient to create constraints to prevent an explosion of possibilities in section 4.4. For any algorithm, fundamentally prioritising any organising information and domain knowledge even at the expense of a strict notion of originality seems to be eventually necessary, and the cost of not doing so to a large enough degree manifests itself as limited applicability or the appearance of noise in the resultant image.

References

Mordvintsev, Olah, Tyka, (2015). 'DeepDream - a code example for visualizing Neural Networks' Google Research

Boiman, Shechtman, Irani, (2008) 'In defense of nearest-neighbor based image classification' In CVPR, Anchorage

Terzic, Mohammed, du Buf, (2015) 'Shape Detection with Nearest Neighbor Contour Fragments' BMVC

Ding, Tong, (2010) 'Content-aware copying and pasting in images' Visual Computer 26: 721.

Zhu, Mumford, (2006) 'A stochastic grammar of images' Found. Trends. Comput. Graph. Vis. 2

Chanda, Dellaert (2004) 'Grammatical methods in computer vision: An overview' Technical Report GIT-GVU-04-29

Belongie, Malik, Puzicha, (2002) 'Shape matching and object recognition using shape contexts' IEEE T-PAMI, 24(4):509–522,

Gatys, Ecker and Bethge, (2016) "Image Style Transfer Using Convolutional Neural Networks," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 2414-2423.

Benny Cheung, (2017) PyDeepStyle [Github Repository] https://github.com/bennycheung/PyDeepStyle.git

Li, Song, Hospedales, Gong (2017) Free-hand sketch synthesis with deformable stroke models Int. J. Comput. Vis., 122 , pp. 169-190

Lateki , Shape data for the MPEG-7 core experiment CE-Shape-1. http://www.cis.temple.edu/~latecki/TestData/mpeg7shapeB.tar.gz

Muja, Lowe (2012) "Fast Matching of Binary Features". Conference on Computer and Robot Vision

Kahaner, Moler, Nash (1988) Numerical Methods and Software. Upper Saddle River, NJ: Prentice Hall.

Liu, Wu, Hsu, Peterson, Xu, (2012) 'On the generation and pruning of skeletons using generalized Voronoi diagrams' Pattern Recognition Letters, Volume 33, Issue 16,