Leveraging Gaussian process approximations for rapid image overlay production

Michael Burke*
Mobile Intelligent Autonomous Systems
Modelling and Digital Sciences
Council for Scientific and Industrial Research
Pretoria, South Africa
mburke@csir.co.za

ABSTRACT

Machine learning models trained using images can be used to generate image overlays by investigating which image areas contribute the most towards model outputs. A common approach used to accomplish this relies on blanking image regions using a sliding window and evaluating the change in model output. Unfortunately, this can be computationally expensive, as it requires numerous model evaluations. This paper shows that a Gaussian process approximation to this blanking approach produces outputs of similar quality, despite requiring significantly fewer model evaluations. This process is illustrated using a user-driven saliency generation problem. Here, pairwise image interest comparisons are used to infer underlying image interest and a Gaussian process model trained to predict the interest value of an image using image features extracted by a convolutional neural network. Interest overlays are generated by evaluating model change at blanking image regions selected using the prediction uncertainty of a Gaussian process regressor.

CCS CONCEPTS

• **Mathematics of computing** → **Probabilistic algorithms**; • **Theory of computation** → **Gaussian processes**; • **Computing methodologies** → **Video summarization**; **Visual content-based indexing and retrieval**;

KEYWORDS

Gaussian processes, Saliency generation

ACM Reference format:
Michael Burke. 2017. Leveraging Gaussian process approximations for rapid image overlay production. In *Proceedings of SAWACMMM'17, Mountain View, CA, USA, October 23, 2017,* 6 pages.
https://doi.org/10.1145/3132711.3132715

*Visiting lecturer in Computer Science and Applied Mathematics, University of the Witwatersrand, Johannesburg, South Africa

SAWACMMM'17, October 23, 2017, Mountain View, CA, USA

ACM ISBN 978-1-4503-5505-6/17/10…$15.00
https://doi.org/10.1145/3132711.3132715

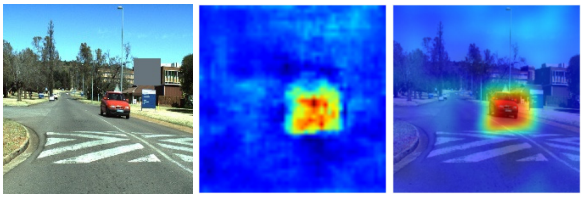


Figure 1: A saliency overlay can be produced by sliding a blanking region (grey) over an image and determining the change in interest prediction output. Smoothing this heatmap produces a saliency overlay highlighting content of interest to an end-user.

1 INTRODUCTION AND RELATED WORK

This work introduces an image interest overlay approximation that can be used to highlight image content of potential interest to an end-user. Saliency or attention overlays of this type can be particularly useful aids for humans tasked with investigating large numbers of images. For example, a medical imaging overlay highlighting image content of potential concern could significantly speed up medical imaging analysis. As a result, the ability to detect and generate saliency images automatically is highly desirable.

Saliency detection algorithms aim to find pronounced features or areas in images and are often used to determine which image areas humans are drawn to. Saliency overlays can also be used to highlight image content of interest to an end-user. The well known Itti-Koch saliency map [10] relies on flagging multiple low level features to build a bottom-up model of image attention. This saliency map has been extended using facial and scene features to highlight images of potential interest to humans in photo albums [21]. Salient image regions have also been extracted using a spectral residual approach [9]. This approach differs from the Itti-Koch saliency map as it is independent of image features, categories and other prior information. Unfortunately, saliency measures such as these often lack contextual information about the type of image content of interest for a specific domain or task.

As a result, feature-based saliency models do not always agree with human definitions of saliency. An attempt to remedy this trains an attention model using a number of hand selected image features by recording human gaze [11]. A support vector machine is then able to classify the potential interest value of an image area using this model. Unfortunately, these approaches aim to build general saliency maps, but for many tasks saliency is domain or problem specific. Contextual information has also been used to combine

low-level features and high-level detections like faces together with visual organisation information to detect salient image areas [7]. Hipster wars [12] trains an image-based style classifier in a fashion application from style judgements, using a part-based model to generate saliency maps that associate clothing items with styles.

Machine learning models are frequently used to make predictions using images. However, these models are often extremely complicated, particularly when deep convolutional neural networks are deployed, and it can be hard to investigate the behaviour of these models. In an attempt to address this, a wide range of model visualisation tools have been developed. Among others, these include t-SNE embeddings [14] that transform image features to make them linearly separable, thereby grouping similar images; retrieving images that maximally activate neural network neurons [6]; investigating neural network filters or visualising the activations and first layer weights [13]. These visualisation tools can often be used to generate model-specific saliency maps.

For example, sensitivity analysis visualisation strategies have been proposed for classification models [3]. These approaches attempt to determine how much a pixel needs to be changed to modify a predicted classification label. An alternative visualisation strategy relies on layer-wise relevance propagation [2]. Here, a relevance score is assigned to each layer of a machine learning model and these relevance scores are propagated through the model to visualise the contributions of single pixels to a model prediction. This approach has been shown to outperform sensitivity analysis visualisation approaches [19] when generating image overlays.

A particularly intuitive model visualisation approach relies on occluding image regions and measuring the change in model output [22]. A large change in model output typically indicates that an image region is important and that the model has been trained to detect regions like this. This behaviour is not only useful for model interpretation, but can be leveraged to generate image overlays for end-users. This approach is similar to sensitivity modelling methods, but a primary benefit is that it allows for black-box models, and does not require knowledge of model structure or access to model layers.

An example of this blanking process being used to generate image overlays can be found in the story-boarding application of [5]. Here, a model is trained to predict the interest value of an image to an end-user for an autonomous mobile robot. This model is then used to build a storyboard summary of a video sequence captured by a mobile robot, and overlays are generated to provide users with a saliency map highlighting image regions likely to be of interest. Unfortunately, this occlusion process can be extremely computationally expensive as it requires numerous model evaluations to complete.

This paper shows that a Gaussian process approximation approach can be used to generate image overlays highlighting image content of interest to an end-user with fewer image blanking evaluations, thereby reducing the processing time required to produce an overlay. The paper is organised as follows. Section 2 briefly describes the interest predictor used to generate overlays in this work, Section 2.1 shows how overlays can be produced using an occlusion approach [22], and Section 2.2 describes the Gaussian process approximation strategy. Finally, results and conclusions are provided in Sections 3 and Sections 4 respectively.

2 GENERATING INTEREST OVERLAYS

For this work, we use a pairwise comparison approach to infer image interest. Here, a user is presented with image pairs, and asked to indicate which image is of greater interest. A Bayesian interest inference algorithm [8] is then applied after a number of image comparisons have been made. Here, images are assumed to have underlying skills or interests, and a probabilistic graphical model of the chance of an image being preferred over another constructed [4]. This model is then used to infer underlying image interests using expectation propagation [17].

Once image interests have been established, a predictive model of interest can be trained. Here, we use a Gaussian process regression model operating on image features extracted using a pre-trained convolution neural network [5].

2.1 Image interest overlays

This section shows how an interest overlay can be created using a predictive model of image interest and a visualisation strategy proposed for convolutional neural networks [22]. Here, the change in algorithm output is observed as a sliding window blanking out image parts is moved over an image. A negative change in output indicates that the blanked image area contained elements of importance.

Given an image interest predictor

$$y = f(\mathbf{X}), \tag{1}$$

where $\mathbf{X}$ denotes an input image, we can generate an overlay by determining the change in image interest predictions as regions in the image are replaced with blanked areas. Let

$$\hat{y}(u, v) = f(\hat{\mathbf{X}}) \tag{2}$$

denote the predictor value at image position (u, v), determined using a new image $\hat{\mathbf{X}}$, formed by replacing the image pixels in a rectangular window around position (u, v) in image $\mathbf{X}$ with the mean pixel values of all images in the dataset for which overlays are being generated.

An image overlay, $\mathbf{I}$, can be generated by determining the resultant change in image interest and taking the exponent of the difference to highlight any differences,

$$\mathbf{I} = \exp(y - \hat{y}). \tag{3}$$

Finally, a Gaussian blur operator is used to smooth the image overlay. Figure 1 illustrates this process.

2.2 Gaussian process overlay approximations

The blanking approach discussed in Section 2.1 is particularly effective at identifying which image content contributed to image interest and generates a useful overlay as a result. However, producing this overlay can be extremely expensive as it typically requires that a feed-forward convolutional neural network evaluation be made for each pixel in the input image. Even if images are downsampled, the large number of evaluations required to generate the overlay is still prohibitively expensive.

This section describes a Gaussian process (GP) approximation strategy that can be used to generate an image overlay using far fewer blanking evaluations. Gaussian processes are collections of

(a) (b) (c)

Figure 2: A Gaussian process regressor is trained to predict image overlay values as a function of image position. The uncertainty in this regression output is evaluated at all image regions and used to propose a new sampling location (red dot (a)). The image is occluded at this location (b), and the change in model output determined by evaluating the predictive model using this blanked image. This value and the corresponding sampling position is added to the Gaussian process training data and the process repeated until convergence (c).

random variables, any finite number of which have joint Gaussian distributions [18].

We can approximate the interest overlay using a Gaussian process $I(\mathbf{x})$, with the mean function,

$$m(\mathbf{x}) = E[I(\mathbf{x})] \tag{4}$$

and covariance function,

$$K(\mathbf{x}, \mathbf{x}') = E[(I(\mathbf{x}) - m(\mathbf{x}))(I(\mathbf{x}') - m(\mathbf{x}'))]. \tag{5}$$

Here, $\mathbf{x}$ corresponds to the rows and columns of pixels in the image. We use a Matérn 5/3 kernel, commonly used for images or grids,

$$K(r) = \sigma \left(1 + \sqrt{5}r + \frac{5}{3}r^2\right) \exp(-\sqrt{5}r), \tag{6}$$

where

$$r = \sqrt{\frac{(u-u')^2}{l_u} + \frac{(v-v')^2}{l_v}}, \tag{7}$$

σ is the kernel variance, (u, u') and (v, v') are the rows and columns of image pixels, and (l_u, l_v) are length scales. Assuming the mean function is zero, we can form a joint normal distribution of training, $\mathbf{I}$, and test, $\mathbf{I}^*$, outputs,

$$\begin{bmatrix} \mathbf{I} \\ \mathbf{I}^* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(\mathbf{x}, \mathbf{x}) & K(\mathbf{x}, \mathbf{x}^*) \\ K(\mathbf{x}^*, \mathbf{x}) & K(\mathbf{x}^*, \mathbf{x}^*) \end{bmatrix}\right). \tag{8}$$

Conditioning the joint Gaussian on the training points, $\mathbf{I}$, produces a predictive overlay distribution,

$$\begin{aligned} \mathbf{I}^*|\mathbf{x}^*, \mathbf{x}, \mathbf{I} \sim \mathcal{N}(K(\mathbf{x}^*, \mathbf{x})K(\mathbf{x}, \mathbf{x})^{-1}\mathbf{I}, \\ K(\mathbf{x}^*, \mathbf{x}^*) - K(\mathbf{x}^*, \mathbf{x})K(\mathbf{x}, \mathbf{x})^{-1}K(\mathbf{x}, \mathbf{x}^*)). \end{aligned} \tag{9}$$

This distribution can be used to estimate interest overlays using a limited number of blanking observations, $\mathbf{I}$. Model fitting is accomplished by initialising the kernel with sensible variance and length scale estimates $(1, 15)$ and then optimising these using a maximum likelihood approach [15].

The probabilistic nature of this overlay prediction is particularly useful, as it allows the construction of an efficient sampling strategy for the selection of blanking observations. We use an incremental sampling strategy, where new samples are chosen by selecting image coordinates with the largest uncertainty in predictive overlay value,

$$\mathbf{x_s} = \arg\max_{\mathbf{x}^*} \left[K(\mathbf{x}^*, \mathbf{x}^*) - K(\mathbf{x}^*, \mathbf{x})K(\mathbf{x}, \mathbf{x})^{-1}K(\mathbf{x}, \mathbf{x}^*)\right]. \tag{10}$$

Figure 2 illustrates this sampling strategy more clearly. This selection process can be slow, but could be bootstrapped using Latin hypercube sampling [16].

3 RESULTS

Empirical Gaussian process overlay approximation convergence was tested on a dataset of 3000 outdoor street scenes captured by an autonomous rover. 15 000 pairwise image comparisons were used to infer image interest using a Bayesian image interest estimation algorithm [5]. Here, TrueSkill [8] is first used to infer image interests from pairwise image comparisons. These interests are then improved using a Gaussian process model that introduces image similarity constraints by applying a squared exponential kernel to image features extracted from dataset images using a pre-trained convolutional neural network [20]. Model specific details can be found in [5], but for this paper it suffices to treat this model as a black-box predictor of image interest.

Attention overlays were generated by sliding a blanking window over each image and evaluating the change in predicted image interest. The proposed Gaussian process overlay approximation algorithm was then tested by measuring the root mean square error (RMSE) in predicted image overlays, as an increasing number of

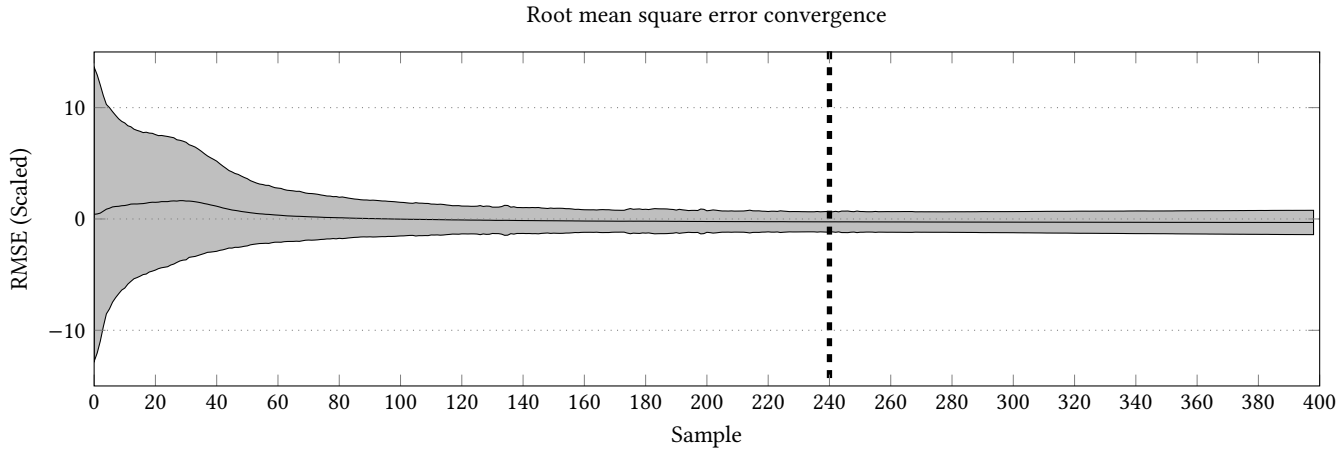


Figure 3: A trace of the scaled root mean square error between approximate and final blanked overlays shows that the error typically converges after about 150 samples. The dashed line shows the time parity point - a 240 sample Gaussian process approximation takes roughly the same amount of time to compute as the full blanked overlay.

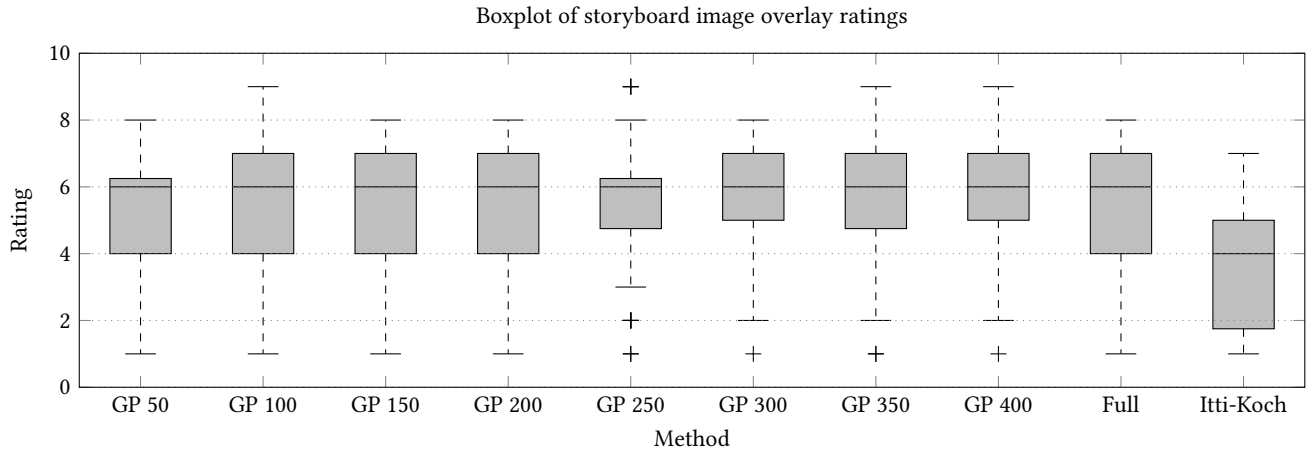


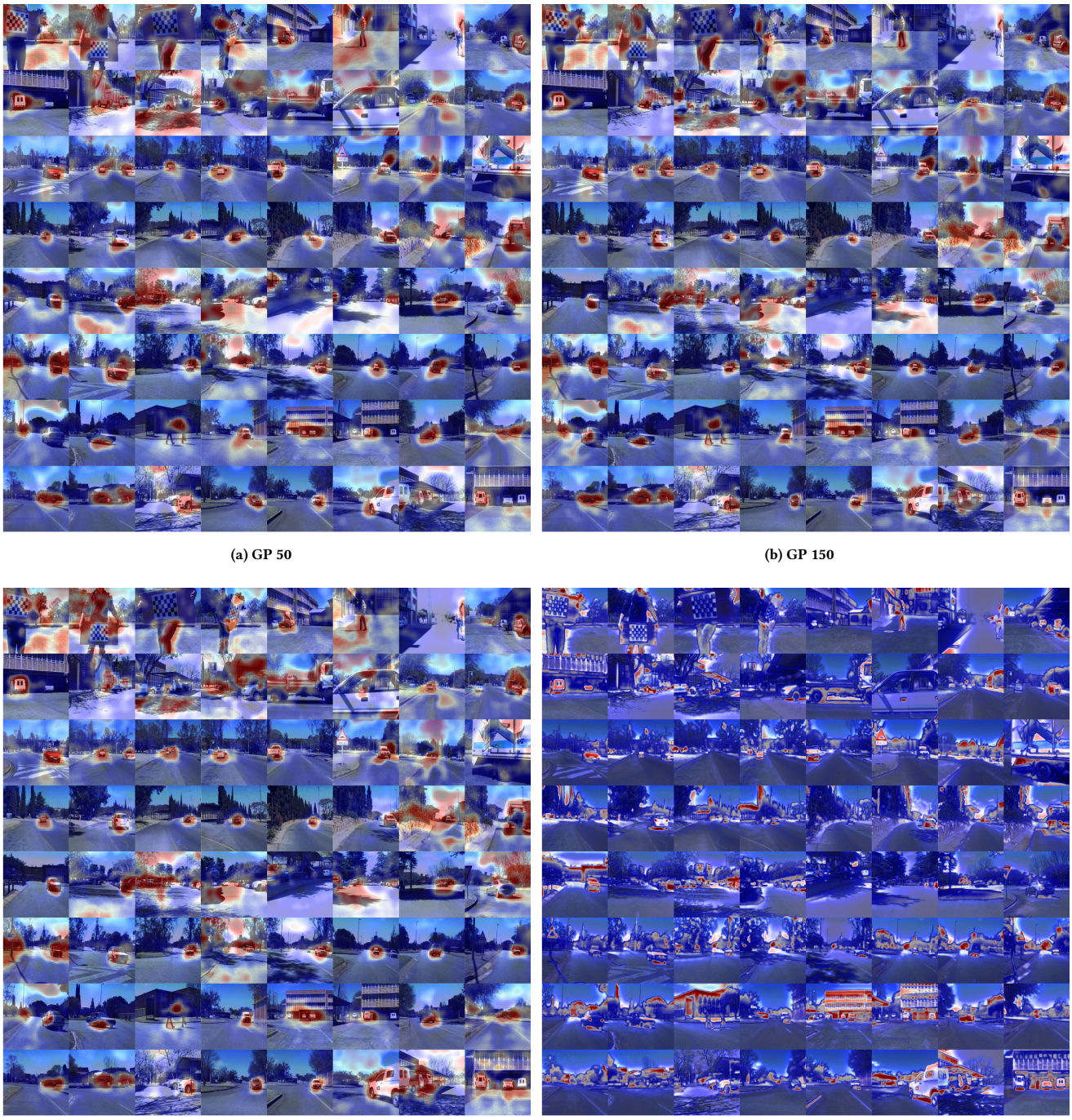
Figure 4: Boxplots of overlay ratings obtained from a domain expert show relatively similar ratings for the approximate and full overlays. This indicates that there is little difference between approximate and full overlays when more than 50 samples are used for overlay approximation. Itti-Koch saliency performs poorly, as it lacks the user context provided by the pairwise comparisons used for image interest inference.

blanked sample measurements were made. The same strides were used for each approach to ensure a fair comparison. GPFlow [15] was used to compute the Gaussian processes.

All experiments were conducted on an Intel Core i7-3930K CPU (3.20GHz x 12) with a GeForce GTX 680/PCIe/SSE2 GPU. Convolutional neural network evaluations were performed in TensorFlow [1] using GPU acceleration.

Figure 3 shows the scaled root mean square errors as an increasing number of blanking evaluations are used to train the Gaussian process model. Shaded traces show the 3-sigma error regions. It is clear that the overlay error seems to converge after roughly 150 samples. The Gaussian process approximation can be trained at approximately 30 samples a second. Individual iterations are slower than convolutional neural network evaluations, but significantly more evaluations are required if a full image overlay is generated. As a result, a full image overlay takes approximately 70 s to generate, which is equivalent to using 240 Gaussian process samples to generate an approximation. This means that the Gaussian process approximation saves approximately 25 seconds per image overlay.

In an attempt to measure the overlay quality, a Likert scale questionnaire was completed for each image in a storyboard generated from the outdoor rover dataset. Here, the 64 most interesting images in the dataset, at least 50 samples apart, were grouped together to form an image summary, complete with saliency overlays. A domain expert was then asked to rate each image in the storyboard using a scale of 1 to 10, with the guidelines that 1 indicated that an



(a) GP 50

(b) GP 150

(c) Full

(d) Itti-Koch

Figure 5: The figure shows 64-image storyboards with saliency maps generated using a 50 sample Gaussian process blanking approximation, a 150 sample Gaussian process blanking approximation, the full blanking process and the Itti-Koch saliency measure.

overlay was not useful, 5 that an overlay partially covered content of interest and 10 that the overlay fully highlighted only content of interest. Figure 4 shows boxplots of the ratings obtained for overlays generated using an increased number of samples to train the Gaussian process approximation. Ratings of overlays produced using the full blanking process and an Itti-Koch saliency generator are also included for comparison.

The boxplots show that there is very little difference in rating distributions for the Gaussian process approximations and the full overlay. This is encouraging, as it means that overlays of equivalent quality can be produced using very few blanking evaluations. As further evidence of this, a Kruskal-Wallis test (H = 2.21, p = 0.97) comparing the Gaussian process approximations and full overlays was unable to reject the null hypothesis that ratings come from the same population distributions. The Itti-Koch saliency performs poorly in the overlay ratings, as it lacks the user context provided by the pairwise comparisons used for image interest inference, and does not provide a useful overlay.

This is also made evident when the actual overlay storyboards generated are inspected (Figure 5). It is clear that the GP approximation produces overlays of similar quality to the full overlay, and successfully highlights image content of interest.

In general, the overlays produced are effective at highlighting smaller vehicles and pedestrians, but struggle with larger content of interest. This could be attributed to the underlying model used to predict image interest, but could also be due to the fixed size (16 pixels) of the blanking window used to generate the image overlay. In practise, this window needs to be tuned to the dataset of interest or varied to produce overlays as different scales.

4 CONCLUSIONS

This paper has shown how a Gaussian process approximation can be used to generate image overlays, requiring fewer image blanking evaluations and significantly speeding up the image generation process as a result. Although we have introduced this approximation in the context of image overlay generation, the proposed approach is also of use for visualising machine learning models and could prove useful for understanding convolutional neural networks.

The experiments conducted here applied an iterative sampling strategy, where image testing points were selected by evaluating the variance of the Gaussian process regressor over image positions. An initialisation strategy using Latin hypercube sampling could improve upon this and reduce the overlay generation time even further.

A user study showed that the approximate overlays were rated similarly to full image overlays. In general, attention overlays are rated highly for smaller objects, but overlays produced for larger objects filling the image are not particularly useful. Future work, which adapts the size of the blanking window as part of the Gaussian process overlay approximation could help to address this.

ACKNOWLEDGEMENTS

This work was supported by a Young Researchers Establishment Grant from the Council For Scientific and Industrial Research, South Africa.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. (2015). http://tensorflow.org/ Software available from tensorflow.org.

[2] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLOS ONE* 10, 7 (07 2015), 1–46. https://doi.org/10.1371/journal.pone.0130140

[3] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert MÄžller. 2010. How to explain individual classification decisions. *Journal of Machine Learning Research* 11, Jun (2010), 1803–1831.

[4] M. Burke. 2016. Image ranking in video sequences using pairwise image comparisons and temporal smoothing. In *2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)*. 1–6. https://doi.org/10.1109/RoboMech.2016.7813166

[5] M. Burke. 2017. User-driven mobile robot storyboarding: Learning image interest and saliency from pairwise image comparisons. *ArXiv e-prints* (June 2017). arXiv:1706.05850

[6] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 580–587.

[7] Stas Goferman, Lihi Zelnik-Manor, and Ayellet Tal. 2012. Context-aware saliency detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34, 10 (2012), 1915–1926.

[8] Ralf Herbrich, Tom Minka, and Thore Graepel. 2007. TrueSkill™: a Bayesian skill rating system. In *Advances in neural information processing systems*. 569–576.

[9] Xiaodi Hou and Liqing Zhang. 2007. Saliency detection: A spectral residual approach. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 1–8.

[10] Laurent Itti and Christof Koch. 2000. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision research* 40, 10 (2000), 1489–1506.

[11] T. Judd, K. Ehinger, F. Durand, and A. Torralba. 2009. Learning to predict where humans look. In *Computer Vision, 2009 IEEE 12th International Conference on*. 2106–2113. https://doi.org/10.1109/ICCV.2009.5459462

[12] M Hadi Kiapour, Kota Yamaguchi, Alexander C Berg, and Tamara L Berg. 2014. Hipster wars: Discovering elements of fashion styles. In *European conference on computer vision*. Springer, 472–488.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[14] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, Nov (2008), 2579–2605.

[15] Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke. Fujii, Alexis Boukouvalas, Pablo Le'on-Villagr'a, Zoubin Ghahramani, and James Hensman. 2016. GPflow: A Gaussian process library using TensorFlow. *arXiv preprint 1610.08733* (Oct. 2016).

[16] M. D. McKay, R. J. Beckman, and W. J. Conover. 1979. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics* 21, 2 (1979), 239–245. http://www.jstor.org/stable/1268522

[17] Thomas Peter Minka. 2001. *A family of algorithms for approximate Bayesian inference*. Ph.D. Dissertation. Massachusetts Institute of Technology.

[18] Carl Edward Rasmussen and Christopher KI Williams. 2006. *Gaussian processes for machine learning*. Vol. 1. MIT press Cambridge.

[19] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. 2017. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems* (2017).

[20] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

[21] Karthikeyan Vaiapury and Mohan S Kankanhalli. 2008. Finding interesting images in albums using attention. *Journal of Multimedia* 3, 4 (2008), 2–13.

[22] Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*. Springer, 818–833.