

University of Ontario Institute of Technology  
Faculty of Science

---

Computer Science

Undergraduate Honours Thesis

# A Framework for Interactive Conference Posters

Candidate:  
Mohammed Ahmed

Thesis advisors:  
Dr. Adam J. Bradley  
Dr. Christopher M. Collins

April 2019

# CONTENTS

1	ABSTRACT	2
2	INTRODUCTION	3
	2.1 Problem Statement and Motivation	3
	2.2 Methodology	3
3	RELATED WORK	5
	3.1 Related Work	5
	3.1.1 VisTiles	5
	3.1.2 Portals for Data Visualization	5
	3.1.3 Toward an interactive poster using digital watermarking and a mobile phone camera	5
4	SYSTEM ARCHITECTURE	6
	4.1 Poster Submission	6
	4.2 Poster Recognition	7
	4.3 Mobile Deployment	9
	4.4 System Evaluation	10
5	CONCLUSION	11
	5.1 Conclusion	11
	5.2 Future Work	11

# 1 | ABSTRACT

In this paper, we present a complete framework for extending conference posters with interactive content. Poster content will become interactive, allowing viewers to manipulate visualizations, and save them for later. We used a web-based backend to allow authors to upload academic posters, and mask areas with interactive content. An image recognition model was trained on submitted posters using the Tensorflow framework [1], and deployed using a mobile application for use at poster sessions.

## 2 | INTRODUCTION

### 2.1 PROBLEM STATEMENT AND MOTIVATION

Poster sessions provide an invaluable opportunity to present work in an informal context. They provide a venue to validate original works, and explore new areas of upcoming research. But, conference posters have an inherent downside - they are not memorable, are not interactive, and their visualizations are limited to presenting data in a single view. This project aims to provide a framework for authors to augment posters with mobile interactivity, and to allow end users to view posters, explore their interactive content, and save them for later. We propose a two part interactive system that consists of: a website for poster submission, a script for recognition model training, and a user facing mobile application for viewers to interact with posters.

### 2.2 METHODOLOGY

This project provides conference participants a method to quickly and efficiently keep track of academic posters they are interested in, and provide them with a meaningful way to interact with static visualizations presented on said posters. This will generate interest in presented posters, and allow authors to present data from multiple views.

Our approach is to create a framework using NodeJS [2] for the poster submission website, Tensorflow [1] for training the image recognition model, and Firebase MLKit [3] for deploying the image recognition model to mobile devices. This framework will allow authors to augment their posters with interactive content, allowing users to view poster information, explore interactive content, and save it for later.

Viewers at the poster session will use the mobile application to interact with the academic posters. As shown in the figure below, users will be presented with a camera view, and a bounding box to place poster in. Providing bounding boxes will encourage the user to capture an image of the poster with limited skew.

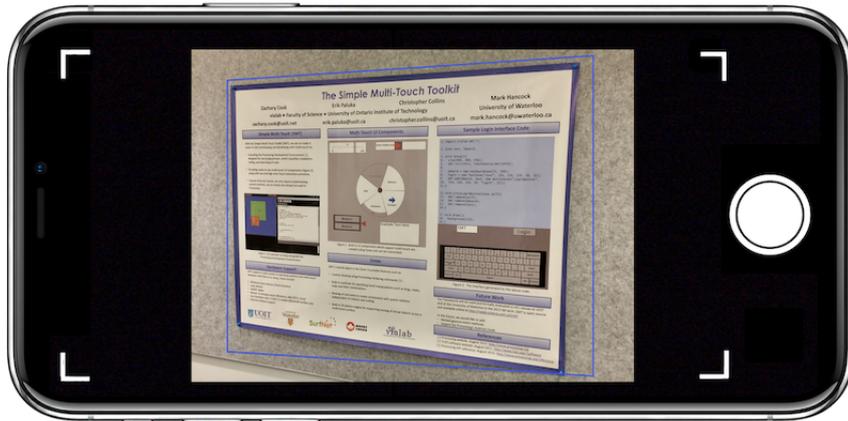


Figure 1: This figure shows a poster being detected by the camera.

After recognizing the poster, the user will be presented with a high-resolution image of the poster captured, along with numbered markers representing areas with interactive content. Upon clicking on one of the markers, the user will be taken to a new screen where they can explore the interactive content. The poster can be saved for later viewing.

While fiduciary markers can be used for poster recognition, we ultimately decided against this as using markers can provide a distraction and consume valuable poster space. This allows the poster to serve its intended purpose for viewers without the our mobile application. By using current deep learning techniques, training a poster recognition model is trivial, and allows us to avoid placing markers on the poster. Using a convolutional neural network gives us the added benefit of allowing the model to recognize posters even when they are partially in frame, and in different lighting conditions.

# 3 | RELATED WORK

## 3.1 RELATED WORK

There has been extensive work in HCI discussing ideas of using mobile devices as a portal or lenses on data visualizations [4], [5], and some work in using fiduciary markers as links [6], but limited work focusing on augmenting posters with interactive content.

### 3.1.1 VisTiles

VisTiles [5] provides a framework for exploring datasets using multiple linked views that are distributed across a set of mobile devices. The web-based prototype runs on a large number of devices and uses an external tracking system, or cross-device pinch gestures to determine the device arrangement. VisTiles showcases the use of mobile devices for exploratory data analysis.

### 3.1.2 Portals for Data Visualization

In this paper by Pahud et al. [4], a set of mobile portals for virtual data visualizations. Using motions sensors to track the current position and orientation of the mobile phone, the proposed mobile experience allows for users to view data through their mobile device like a portal into a visualization. 3D bookmarking is used to store the virtual location of the user for later retrieval.

### 3.1.3 Toward an interactive poster using digital watermarking and a mobile phone camera

This paper by Pramila et al. [6] outlines a method for embedding a watermark in printed media with a camera phone. The work mainly focuses on posters used for print advertisements. Using an embedded watermark, the proposed system detects the watermark on posters, and forwards the user to a web link.

# 4 | SYSTEM ARCHITECTURE

The system architecture is comprised of three main components: a website for poster submission, a script for recognition model training, and a user-facing mobile application for viewers to interact with posters. Each of these components will be explored in depth in the sections that follow.

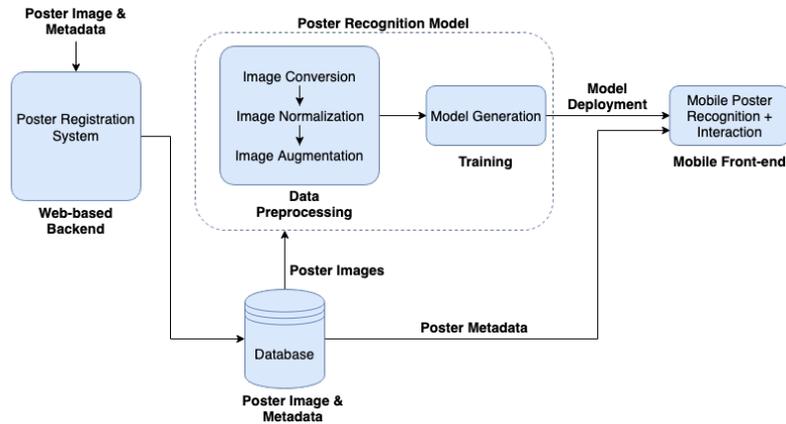


Figure 2: This figure outlines the various components in the system architecture diagram.

## 4.1 POSTER SUBMISSION

The first step to making printed posters interactive is creating a dataset from posters we want to recognize, and their respective metadata. A poster submission website was chosen for this task for ease of data collection. In order to submit a poster, the author must first register for an upcoming event. This can be done by clicking on the *Find Conference* button on the navbar to the left, as shown in the following figure. The website was implemented using NodeJS [2] as the backend, React [7] for the front-end components, and Firebase Cloud Firestore [8] as the database of choice for its NoSQL structure, and robustness.

After registering for a conference, the event will appear on the left under *Upcoming Events*. Clicking on a registered upcoming event will present the user with a poster submission page where they are required to submit a high-quality copy of their poster, along with its metadata including: the title of the poster, all contributing authors and emails, and the type of interactive content they wish to provide. Currently, interactive content is limited to videos and web links. Numbered markers for the interactive content locations are provided as an overlay on the submitted poster image and should be moved to their appropriate location.

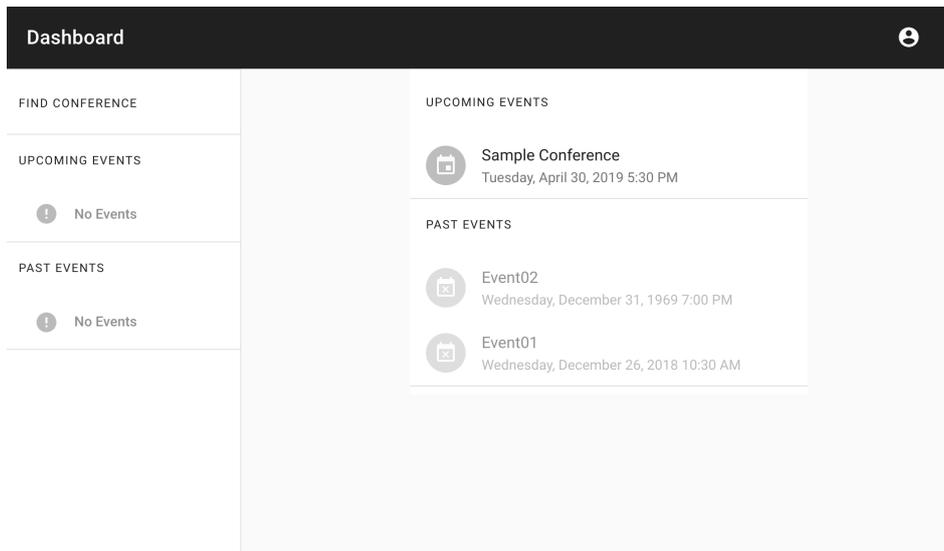


Figure 3: This figure shows the conference registration page. It provides a list of upcoming events that the user can register for.

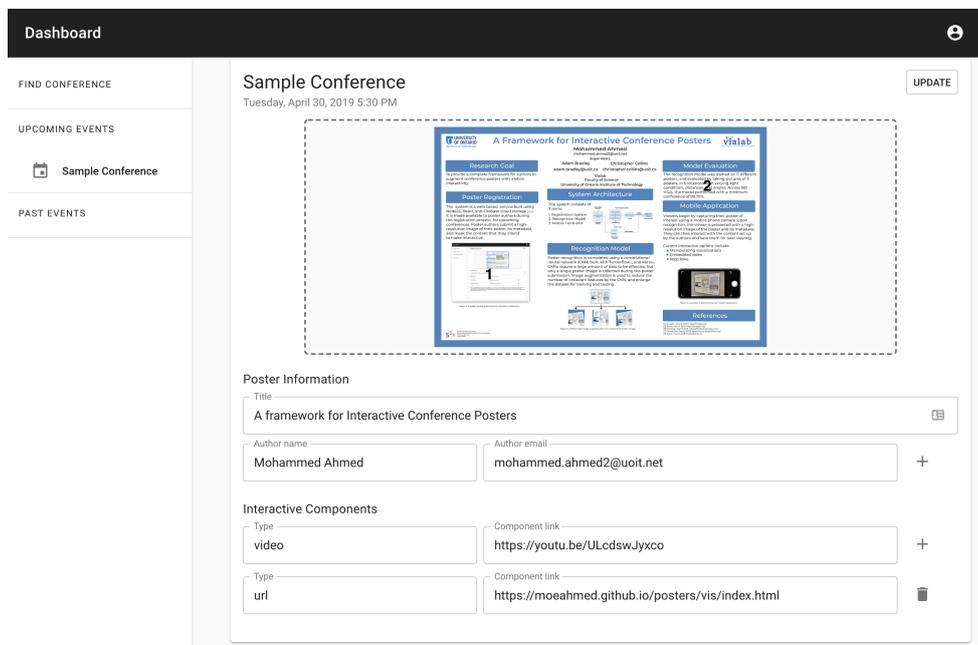


Figure 4: This figure shows the poster submission page. As shown, the user has submitted a poster along with its metadata and marked areas intended for interactive content.

Upon submission, the poster image and its respective metadata is stored in a database for later retrieval in the poster recognition model training, and mobile deployment phases.

## 4.2 POSTER RECOGNITION

After collecting the posters using the poster submission website, we can begin training a poster recognition model to identify individual posters. A

training script was developed in Python using the TensorFlow framework [1]. A convolutional neural network (CNN) was used for poster image recognition. Given the limited hardware resources, a small CNN architecture is required. A custom CNN model was used inspired by MobileNet V1 [9] was used due to the limited hardware resources of mobile devices. While MobileNet performs on par with commonly used CNN model such as VGG16 [10] and GoogleNet [11], it’s model size is much smaller and more efficient. In most image detection applications, retraining an existing model like MobileNet to fit your use case would be sufficient, but in this use case we need to perform poster rather than detection. As such, we require the model to recognize the specified poster rather than determine if it is of the class poster. This requires a higher input image resolution to allow the CNN to detect features in the poster images. The CNN input image size was set to 400x400 pixels.

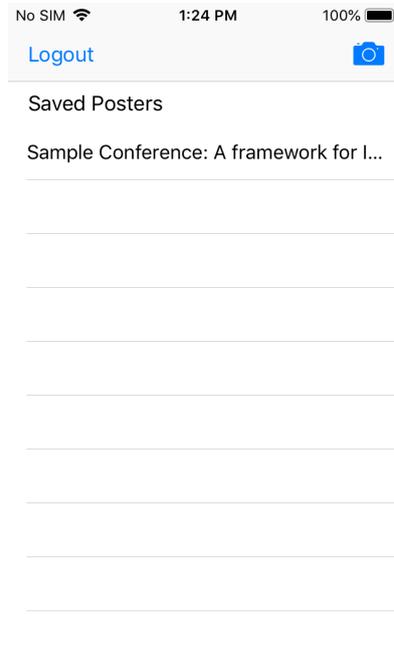
Layer Type/Stride	Filter Shape	Input Size	Activation Function
Input	1	400 x 400 x 3	-
Conv /s1	3 x 3 x 3 x 32	400 x 400 x 32	Relu
Max Pooling /s2	2 x 2	400 x 400 x 32	Relu
3x Conv /s1	3 x 3 x 3 x 64	200 x 200 x 64	Relu
Max Pooling /s2	2 x 2	200 x 200 x 64	Relu
3x Conv /s1	3 x 3 x 3 x 128	100 x 100 x 128	Relu
Max Pooling /s2	2 x 2	100 x 100 x 128	Relu
2x Conv /s1	3 x 3 x 3 x 256	50 x 50 x 256	Relu
Max Pooling /s2	2 x 2	50 x 50 x 256	Relu
Conv /s1	3 x 3 x 3 x 512	25 x 25 x 512	Relu
Dense	-	100	Relu
Dense	-	100	Relu
Dense	-	100	Relu
Output	-	11	Softmax

**Table 1:** The poster recognition model’s convolutional neural network structure.

Recognition tasks require a large amount of data to train and be effective, and the poster submissions only provide a single image of each poster. In order to create a dataset of sufficient size for training, testing, and validation, we must perform image augmentation. Poster images were augmented using positional, and colour shifting. This is done to mimic the images passed to the poster recognition model by the mobile application. This causes the model to overfit on the poster images provided, but in this case overfitting is required because we aim to recognize individual posters rather than classify whether or not the image is of a poster. Poster images and their respective metadata are pulled from the database by their event ID. Each poster is resized to 400x400 pixels before augmenting. Each poster is represented as a unique class by it’s unique submission ID. After training the model, the model is converted to a TFLite model for mobile deployment. The TFLite model is then uploaded to the cloud using Firebase ML Kit for ease of distribution and maintainability.

## 4.3 MOBILE DEPLOYMENT

The mobile application is used for interacting with posters at registered events. After creating an account, and signing in, the user is presented with the screen shown in the following figure. Saved posters can be found in the list presented on the main screen. Clicking on a saved poster will show the poster and it's interactive content represented by numbered markers.



**Figure 5:** This figure shows the main screen from the mobile application. Saved posters can be found in the list shown.

Upon clicking on the camera, users can begin by capturing their poster of interest. A bounding box will appear over any detected posters. After clicking the capture button, the device will crop the image and perform perspective correction resulting in an image of the poster with no background. This is done to reduce the noise in the input image before passing it to the poster recognition model. This greatly increases the accuracy of the recognition model as the input image will very closely resemble images from the training set, and is required due to the model overfitting performed earlier, and the loss of layers in the CNN when converting the model to a TFLite model. The image is then sent to the cloud where recognition model is stored.

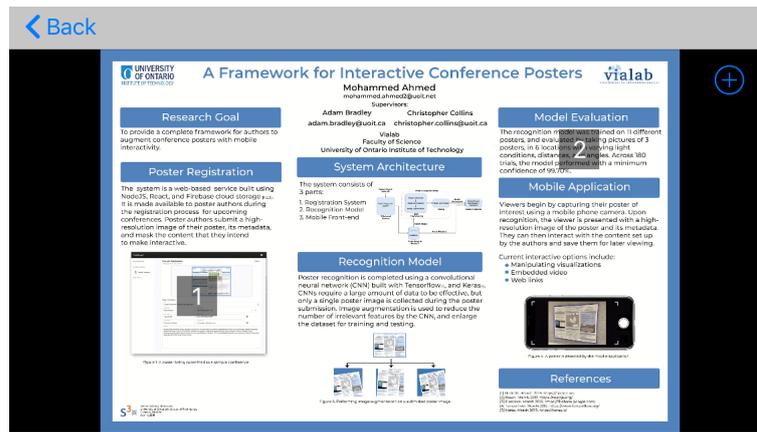


Figure 6: This figure shows the preview screen from the mobile application. Interactive content is represented by the numbered markers.

After recognizing the poster, the user is presented with a high resolution image of the poster overlaid by numbered markers representing the interactive areas on the poster. They can then interact with the content presented, and save it for later viewing by clicking on the save button on the top right corner.

#### 4.4 SYSTEM EVALUATION

The poster recognition model was trained on 11 different posters, and evaluated by taking pictures of 3 posters, in 6 locations with varying light conditions, distances, and angles. Across 180 trials, the model performed with a minimum confidence of 99.70%. This validates the performance of the recognition model. While the high accuracy of the model can be directly attributed to the removal of background elements in the input image by cropping and perspective correction, it was necessary to improve the accuracy of the recognition model due to the limitation of only being able to acquire a single image of each poster for training. Performing perspective correction and cropping the background of the input image will result in an image very similar to those present in the training set.

# 5 | CONCLUSION

## 5.1 CONCLUSION

We presented a framework for authors to augment posters with mobile interactivity, administrators to train posters for conferences, and to allow end users to view posters, explore their interactive content, and save them for later. The two part interactive system that begins with a website allowing authors to upload academic posters and mask areas with interactive content, and a user facing mobile application for viewers to interact with posters. This framework provides poster authors the flexibility of augmenting their posters with online visualizations, video, and web content. This gives viewers an exciting, memorable way to interact with posters and explore them in depth at any time.

## 5.2 FUTURE WORK

While we believe the framework is an innovative method of interacting with static posters, the functionality is still limited. We plan on extending the framework to accept more interactive content types, allowing authors more flexibility in content they can augment posters with, and provide viewers with an memorable experience. we wish to allow users to annotate, comment, and share posters with others.

With the abundance of posters stored in the database, we believe it would be notable to extend the framework to recommend posters of potential interest to users, and implement user profiles containing published posters that can be explored by all. We believe this will encourage collaboration, and draw on the social nature of poster sessions, and allow users to start discussions on future work.

## REFERENCES

- [1] Martin 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, Y. 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, *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015. [Online]. Available: <http://tensorflow.org/>.
- [2] NodeJS. (2019). Node.js, [Online]. Available: <https://nodejs.org/en/> (visited on 03/14/2019).
- [3] Firebase. (2019). ML kit for firebase | machine learning for mobile developers, [Online]. Available: <https://firebase.google.com/products/ml-kit/> (visited on 03/16/2019).
- [4] M. Pahud, E. Ofek, N. H. Riche, C. Hurter, and J. Grubert, “Mobiles as portals for interacting with virtual data visualizations”, *CoRR*, vol. abs/1804.03211, 2018. arXiv: [1804.03211](https://arxiv.org/abs/1804.03211). [Online]. Available: <http://arxiv.org/abs/1804.03211>.
- [5] R. Langner, T. Horak, and R. Dachsel, “Demonstrating vistiles: Visual data exploration using mobile devices”, in *Proceedings of the 2018 International Conference on Advanced Visual Interfaces*, ser. AVI '18, Castiglione della Pescaia, Grosseto, Italy: ACM, 2018, 69:1–69:3. DOI: [10.1145/3206505.3206583](https://doi.org/10.1145/3206505.3206583).
- [6] A. Pramila, A. Keskinarkaus, and T. Seppänen, “Toward an interactive poster using digital watermarking and a mobile phone camera”, *Signal, Image and Video Processing*, vol. 6, pp. 211–222, 2012.
- [7] ReactJS. (2019). React - a javascript library for building user interfaces, [Online]. Available: <https://reactjs.org/> (visited on 03/14/2019).
- [8] Firebase. (2019). Cloud firestore | store and sync app data at global scale, [Online]. Available: <https://firebase.google.com/products/firestore/> (visited on 03/14/2019).
- [9] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications”, *CoRR*, vol. abs/1704.04861, 2017. arXiv: [1704.04861](https://arxiv.org/abs/1704.04861). [Online]. Available: <http://arxiv.org/abs/1704.04861>.
- [10] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, *arXiv preprint arXiv:1409.1556*, 2014.

- [11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions", *CoRR*, vol. abs/1409.4842, 2014. arXiv: [1409.4842](https://arxiv.org/abs/1409.4842). [Online]. Available: <http://arxiv.org/abs/1409.4842>.