

arXiv:2105.14184v1 [ssCV] 29 May 2021

# E2ETag: An End-to-End Trainable Method for Generating and Detecting Fiducial Markers

Brennan Peace  
[jpeace@huskers.unl.edu](mailto:jpeace@huskers.unl.edu)  
 Eric Psota  
[epsota@unl.edu](mailto:epsota@unl.edu)  
 Yanfeng Liu  
[yanfeng.liu@huskers.unl.edu](mailto:yanfeng.liu@huskers.unl.edu)  
 Lance C. Pérez  
[lperez@unl.edu](mailto:lperez@unl.edu)

University of Nebraska-Lincoln  
 Lincoln, Nebraska 68588-0511

## Abstract

Existing fiducial markers solutions are designed for efficient detection and decoding, however, their ability to stand out in natural environments is difficult to infer from relatively limited analysis. Furthermore, worsening performance in challenging image capture scenarios - such as poor exposure, motion blur, and off-axis viewing - sheds light on their limitations. E2ETag introduces an end-to-end trainable method for designing fiducial markers and a complimentary detector. By introducing back-propagatable marker augmentation and superimposition into training, the method learns to generate markers that can be detected and classified in challenging real-world environments using a fully convolutional detector network. Results demonstrate that E2ETag outperforms existing methods in ideal conditions and performs much better in the presence of motion blur, contrast fluctuations, noise, and off-axis viewing angles. Source code and trained models are available at <https://github.com/jbpeace/E2ETag>.

## 1 Introduction

Visual tracking aims to locate targets as they move through the field of view, while maintaining a consistent identification as targets disappear, reappear, and change their appearance [1]. The identity assigned to targets is, in general, arbitrary and their exact location is often represented via a bounding box [2] or a collection of key points [3, 4].

To obtain the precise location, identity, and pose of targets, one can use fiducial markers, which are man-made objects designed to be placed in (augment) a scene. Along with algorithms to detect and classify them, they provide a plug-and-play tracking method that is scene-agnostic. Perhaps the most well-known fiducial markers are QR codes. When placed conveniently in front of a camera, QR codes can be detected and decoded efficiently [5]. They are so ubiquitous that most modern cell-phone camera applications instantly recognize and decode them by default. QR codes are capable of encoding thousands of bits of information, but they are not designed to overcome difficult viewing conditions.

This work targets a sub-category of fiducial markers aimed at challenging image capture scenarios. Within this category, several methods have been proposed by the research community [6, 7, 8]. Nearly all of them use two-dimensional bit encoding and heuristically

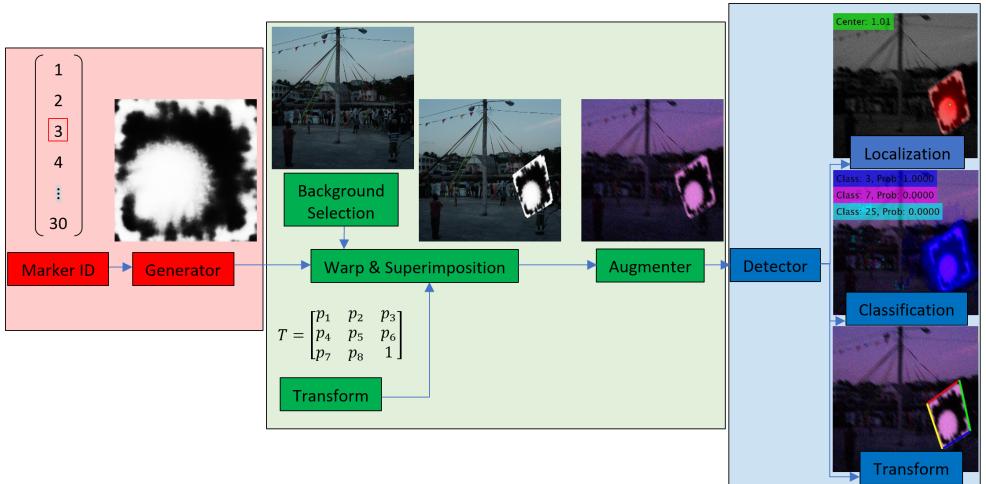


Figure 1: Simplified model flowchart of E2ETag used to generate and detect fiducial markers.

designed detectors. While computationally efficient and reliable against false detections, they are not explicitly designed to handle real-world challenges like motion blur and noise.

The fiducial marker method introduced here takes a machine-learnable approach to both marker generation and detection. It relies on three stages that are end-to-end trainable, as illustrated in Figure 1. The first stage generates the fiducial marker from a one-hot vector using a transposed convolution. The second stage randomly augments the marker and superimposes it into a real image. Finally, the third stage uses a fully-convolutional network to detect the location, identity, and pose of the marker. The detector and generator learn to adapt to severe augmentations and differentiate the marker from objects in real-world environments.

The contributions of this work include:

- An end-to-end trainable framework for generating and detecting fiducial markers.
- Randomized, backpropagatable superimposition for simulated image capture.
- Analysis on the effectiveness of synthetic training for real-world applications.
- A methodology for evaluating the robustness of fiducial markers.

## 2 Related Work

Fiducial markers are designed to stand out from the environment and achieve a high detection rate while being, at the same time, distinguishable from one another for multi-tag detection. Traditionally, designs use a pre-determined library of decodable square patterns. ARToolKit [10], one of the earliest fiducial markers, features an arbitrary pattern enclosed by a black border (Figure 2(a)). Performance is limited by the number of patterns and the camera resolution. The arbitrary nature of its content also makes inter-tag classification difficult to guarantee. ARTag [9] proposed to fix this by using binary block patterns and introducing error-correction coding into its design (Figure 2(b)).

RuneTag [8] (Figure 2(c)) exploits the projective properties of circular dot patterns and error-correction coding. The dots form one or more concentric circles, and the fact that both

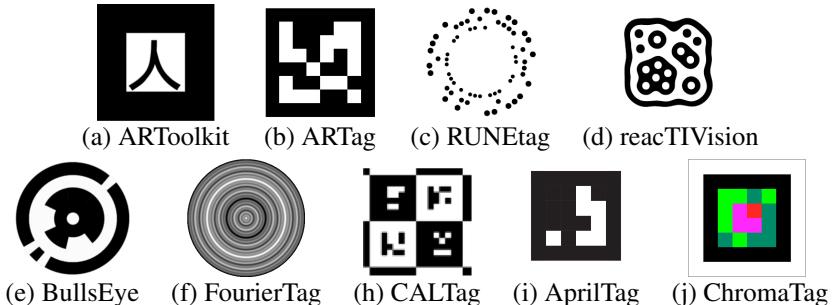


Figure 2: Examples of existing fiducial marker designs.

rings and dots appear elliptical under projective transformation makes decoding straightforward. This design is robust under partial occlusion, blur, and noise. The authors of reacTIVision [2] proposed topology-based irregular shapes for fast detection (Figure 2(d)). It supports a large number of markers, with a size that changes based on the number of encoded features. The design was originally proposed for table-based musical instruments [10] but can serve as a general-purpose marker. BullsEye is another topological pattern targeting the same applications as reacTIVision [13] (Figure 2(e)). It improves upon the precision of reacTIVision and introduces GPU-enabled detection, however, both of these methods target two-dimensional location and orientation and require multiple markers for pose estimation.

FourierTag [20] encodes information in the amplitude of a marker’s Fourier transform (Figure 2(f)). It is designed to gradually degrade the quality of encoded information as distance increases and/or viewing angles worsen, instead of being unrecognizable abruptly. The high-order bits are encoded with low frequencies and low-order bits with high frequencies. As a result, the encoded bits have variable length depending on the viewing distance.

CALTag [11] (Figure 2(h)) proposes a high-density marker design as an alternative to checkerboard patterns and uniquely identifiable markers, with the application of camera calibration in mind instead of augmented reality. It offers an automatic processing procedure without parameter fine-tuning, which benefits multi-camera applications.

AprilTag [12, 19] was designed to improve upon ARTag (Figure 2(i)). It proposes a grid of black and white blocks that serve as a binary payload with guaranteed minimum Hamming distance between markers undergoing 0, 90, 180, and 270 degree rotations. It was originally designed to handle partial-occlusion recovery, but the authors concluded that occluded markers were rarely useful and they instead targeted detection and decoding speed.

ChromaTag [6] features adjacent red and green blocks surrounded by black and white rings (Figure 2(j)). The red and green pattern is rare in natural scenes and reduces initial false detections. The black and white rings provide high contrast for localization. In the CIELAB color space, each color in the design has a consistent value in the A channel and a different value in the B channel, making the design easy to detect and decode.

Existing designs use hand-crafted patterns and detection algorithms. It is unclear if detection is optimized for the marker or vice versa. To the best of our knowledge, the method introduced in this paper is the first end-to-end trainable fiducial marker solution. The marker designs are jointly optimized with their detector, allowing for designs that learn to stand out from the environment while simultaneously learning to look different than each other.

### 3 Method

The model used during training is composed of a three-stage generator/augmenter/detector network (Figure 1). The first stage generates markers through a transposed convolution.

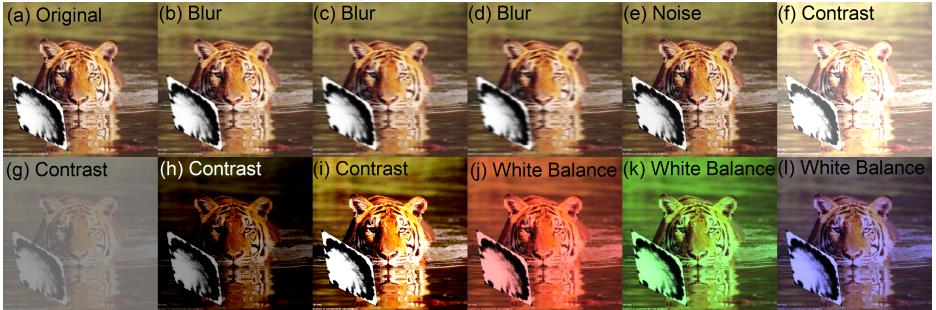


Figure 3: Superimposed marker without augmentations (Figure3a). Motion Blur at random angles with varying kernel length 5, 10, and 15 (Figure3 b,c,d). Additive noise ranging (-0.15,0.15) (Figure3e). Contrast with  $W = 1.4$  and  $B = 0.4$  (Figure3f),  $W = 0.6$  and  $B = 0.4$  (Figure3g),  $W = 0.6$  and  $B = -0.4$ , and  $W = 1.4$  and  $B = -0.4$  (Figure3i). White balance (1.3,0.7,0.7) (Figure3j), (0.7,1.3,0.7) (Figure3k) and (0.7,0.7,1.3) (Figure3l).

The second stage warps them onto a sample image and applies augmentations that simulate real-world image capture. The final stage estimates the marker’s location, class, and pose.

### 3.1 Generator

The generator is defined by a transposed convolution where the input is a one-hot vector indicating the desired class. The transposed convolution consists of a single kernel of size  $S \times S \times C$ , where  $S = 128$  and  $C = 30$ . When the  $1 \times 1 \times C$  one-hot vector is convolved with the kernel, the  $S \times S \times 1$  output is a single kernel layer representing an E2ETag image.

### 3.2 Spatial Warp and Superimposition

The E2ETag image is warped and superimposed into a real image. Background images are randomly sampled from the COCO [1] and Imagenet [2] data sets and resized to  $640 \times 640$ . The spatial warping transform is constructed by randomly generating  $x$  and  $y$  translations  $\{t_x, t_y\}$  ranging  $(0, 640)$ , rotation  $r$  ranging  $(0, 2\pi]$ , scaling  $\{s_x, s_y\}$  ranging  $(8/128, 320/128)$ , shear  $\{h_x, h_y\}$  ranging  $(-3\pi/12, 3\pi/12)$ , and projective warping  $\{w_x, w_y\}$  ranging from  $(-0.0015, 0.0015)$ . Additionally, a shift of  $-S/2$  is applied to both dimensions to zero center the marker at the origin prior to warping. The resulting projective matrix is

$$T = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos(r) & -\sin(r) & 0 \\ \sin(r) & \cos(r) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & h_y & 0 \\ h_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & -S/2 \\ 0 & 1 & -S/2 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \\ p_7 & p_8 & 1 \end{bmatrix}. \quad (1)$$

The parameters corresponding to rotation, scaling, and shearing ( $p_1, p_2, p_4, p_5, p_7, p_8$ ) are used by the loss function for training, as discussed in section 3.5.

### 3.3 Local and Pixel-Level Augmentations

Four separate augmentations are applied sequentially to images with superimposed markers: motion blur, white-balance, contrast, and additive noise, as illustrated in Figure 3. Motion blur is simulated first by convolving a blur kernel with variable angle and length. This kernel simulates linear camera motion along a direction  $d$  uniformly ranging  $(0, 2\pi]$  with pixel length  $l$  uniformly ranging  $(0, 10)$  pixels. Input Image  $I$  is convolved with kernel  $\phi(d, l)$  to produce the output image  $I_{MB} = I * \phi(d, l)$ .

White balance simulates lighting conditions with varied temperatures. Random values for each channel, uniformly ranging  $(0.7, 1.3)$ , scale each channel of an RGB input image.

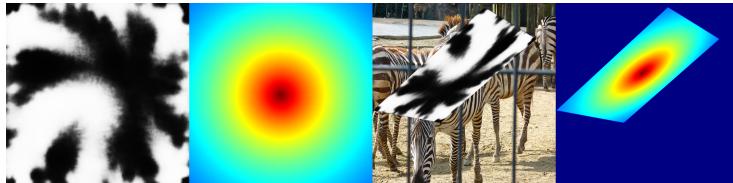


Figure 4: Sample E2ETag and decaying exponential used to encode marker location. Both marker and location encoding are superimposed the same way into the image. In the location encoding, dark red indicates a value of 1.0 and dark blue indicates a value of 0.0.

The resulting image is  $I_{WB} = I_{MB} \odot C$ , where  $C$  is an image of the same size as  $I_{MB}$  where each channel is given a single scale value and  $\odot$  is an element-wise product.

A contrast augmentation is then used to simulate variations in exposure, post processing, and light pollution. Contrast considers two random variables: the maximum white  $W$  uniformly distributed  $(0.6, 1.4)$  and minimum black  $B$  uniformly ranging  $(-0.4, 0.4)$ . Contrast is adjusted using  $I_C = I_{WB} \times (W - B) + B$ .

Finally, a noise image  $N$  with pixel values uniform in the range  $(-n/2, n/2)$  is added to the input image. Image  $I_C$  is augmented with noise using  $I_N = I_C + N$ . Finally, pixel values in  $I_N$  are clipped between 0 and 1.

### 3.4 Detector

The final stage of the model, used for both training and testing, is the detector. The detector localizes the marker within the image, classifies it, and estimates the transformation parameters. The pre-trained, fully-convolutional network chosen in this work is the DeepLabV3+ architecture [■] with a ResNet18 core [□]. This network was chosen due to its speed, classification performance, ability to handle multi-scale detection, and high resolution feature space. Instead of up-sampling back to the input resolution, the output before the first transposed convolution is used and the resulting network down-samples the input image scale from  $640 \times 640$  to  $80 \times 80$ . A  $1 \times 1 \times 256 \times 256$  convolution/batchnorm/ReLU block and  $1 \times 1 \times 256 \times 37$  convolution layer are added to provide an output with the number of channels used to encode targets. Those channel encodings are described in the following.

#### 3.4.1 Detector Channel Encoding

The first channel, of size  $80 \times 80 \times 1$ , is used to detect and localize the marker. The marker’s location in the image is encoded via a decaying exponential setting the value at each pixel to  $e^{-(r/64)}$ , where  $r$  is the distance of each pixel from the center. The decaying exponential image has the same size as the marker with dimensions  $128 \times 128$ . Figure 4 illustrates a marker image and its corresponding mapping in the first channel before and after warping and superimposing. To detect tag locations, the method begins by finding all regional maxima in  $3 \times 3$  regions in the  $80 \times 80 \times 1$  output. If the value of the regional maxima exceeds 0.5, it is considered a tag center and its sub-pixel peak is estimated using quadratic interpolation.

The next 30 channels, with feature space size  $80 \times 80 \times 30$ , encode marker identities using softmax pixel-wise classification. Classification is trained equally for all pixels occupied by the transformed marker. All other grid locations are allowed to choose identities without affecting the loss, so that uncertainty in detection does not affect classification.

The last six channels of the output, with feature space size  $80 \times 80 \times 6$ , contain the projective parameters  $(p_1, p_2, p_4, p_5, p_7, p_8)$  at each pixel location occupied by the marker.

### 3.5 Training Details

The network is end-to-end trainable through the detector, augmentation, and warp layers. While improving the detector, training also encourages marker designs that are easy to detect

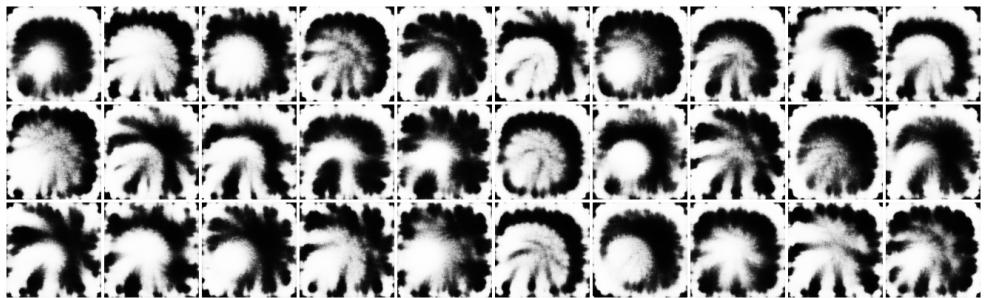


Figure 5: E2ETag markers generated with 30 classes.

and classify. While the detector was mostly pre-trained, the transposed convolution weights used to generate markers were randomly initialized with Gaussian samples that have mean 0.0 and variance 1.0. The bias was fixed to zero. Outputs of the transposed convolution are passed through a preliminary contrast augmentation layer then to a sigmoid layer to limit values between 0 and 1 prior to superimposing into images. Because early layers typically train much more slowly than later layers, the learning rate of the transposed convolution was increased by a factor of 1000. The Adam optimizer was used with a learning rate of  $2 \times 10^{-5}$ , a batch size of 8, and an L2 regularization of  $10^{-4}$ . The complete set of markers used in evaluation are depicted in Figure 5.

### 3.6 Backward Propagation Through Warping and Augmentation

When superimposing markers, transformations  $T$  were accepted only if each marker corner remained inside the image bounds. For back-propagation, the inverse transformation  $T^{-1}$  was used to map gradients back to their origins in the marker image. The inverse transform also ensures that gradients are only applied to the marker; background regions are ignored. Both forward and backward transformations use bilinear interpolation.

Gradients passed through the motion blur operation are back-propagated by reusing the motion blur kernel on the gradients. Gradients for white balance and contrast adjustments are scaled according to their multiplicative factors.

### 3.7 Loss Function

Loss is the aggregate penalty of errors in classification, localization, and estimation of transform parameters. Detection/localization loss,  $L_{\text{loc}}$ , is penalized as the mean-square error, given by

$$L_{\text{loc}} = \sum_{i=1}^{\text{Rows}} \sum_{j=1}^{\text{Cols}} (y_{\text{loc}}(i, j) - \hat{y}_{\text{loc}}(i, j))^2 / \text{NumPix}, \quad (2)$$

where  $y_{\text{loc}}$  is the image target localizations,  $\hat{y}_{\text{loc}}$  is the image with predicted localizations, and NumPix is the number of pixels within the warped marker region.

Classification loss is the categorical cross-entropy loss of target class versus the predicted class, given by

$$L_{\text{class}} = - \sum_{c=1}^C \sum_{i=1}^{\text{Rows}} \sum_{j=1}^{\text{Cols}} y_{\text{class}}(i, j, c) \odot \log(\max(\hat{y}_{\text{class}}(i, j, c), \epsilon)) / (C \cdot \text{NumPix}), \quad (3)$$

where  $C$  is the number of classes,  $y_{\text{class}}$  is the image target classifications,  $\hat{y}_{\text{class}}$  is the image with predicted classifications, and  $\epsilon = 10^{-9}$  is a constant used to prevent division by zero.

The projective transformation parameters are not penalized directly; instead, the corners of the target marker and the predicted marker are calculated from the estimated projective parameters. Parameters  $p_3$  and  $p_6$  are set to zero to isolate transformation from localization. The predicted corners  $\hat{\mathbf{c}}$  and target corners  $\mathbf{c}$  are both normalized by the standard deviation of the target corners  $\sigma_c$  to produce  $\mathbf{c}_N$  and  $\hat{\mathbf{c}}_N$ . Normalizing the corners allows projective transforms to be penalized with scale invariance, preventing large markers from dominating training. The  $K$  strongest target localizations in  $y_{loc}$  determined the grid locations to sample the transformations. Here, the strongest  $K = 5$  detections were chosen to sample the true center and its four abutting neighbors. Projective transformation loss is the mean-absolute error between normalized predicted and target corners, defined by

$$L_{proj} = \sum_{k=1}^K \sum_{i=1}^4 \sum_{j=1}^2 |\hat{\mathbf{c}}_N(i, j, k) - \mathbf{c}_N(i, j, k)| / (8 \cdot k), \quad (4)$$

where  $j$  is the index for an  $(x, y)$  coordinate,  $i$  is the index for the corner, and  $k$  is the index for each localization maxima. Finally, the total loss  $L$  is an aggregate of all losses, given by

$$L = a \cdot L_{class} + b \cdot L_{loc} + L_{proj}, \quad (5)$$

where  $a = 100$  and  $b = 50$  are scalar constants derived empirically to balance training.

## 4 Results

E2ETag is compared to two state-of-the-art fiducial marker methods: ChromaTag and AprilTag. Two different metrics are used to evaluate performance: detection and classification. For detection accuracy, the intersection over union (IoU) derived from the corner locations is used and a true positive detection is defined by IoU greater than 50% with the ground truth. Classification performance is simply evaluated as correct or incorrect class output.

### 4.1 Data Collection

For each image used in the evaluation, a single ChromaTag, AprilTag, and E2ETag are placed in a scene with similar pose. Each of these methods was configured to support 30 different marker classes (AprilTag and ChromaTag use 16H5 encoding). Images with all three markers in each frame are captured at  $3024 \times 3024$  resolution with a 35mm equivalent focal length of 52mm. They are captured in seven different environments, placed in varying lighting environments, and mounted on man-made structures as well as natural environments.

The markers were attached to a cardboard square,  $7.62 \times 7.62$  cm, and mounted to different surfaces including trees, poles, a stone structure, a reflective glass window, and a chain-link fence. Seven different markers were used with identities 5, 6, 7, 9, 11, 19, and 27. Twenty-five images from each environment were captured at five different angles ( $-80^\circ$ ,  $-40^\circ$ ,  $0^\circ$ ,  $40^\circ$ ,  $80^\circ$ ) and five different distances (1, 2, 3, 4, and 5 meters). The true corner locations were hand-annotated for each marker at the original resolution  $3024 \times 3024$  and downsampled to  $640 \times 640$ .

### 4.2 Performance Comparison

The chosen metric for localization accuracy is the IOU of the quadrilaterals defined by the corners of the predicted and annotated corners. Recall and precision are used to evaluate the performance of localization. True positives require 50% IoU with the ground truth quadrilaterals. False negatives are defined by an undetected ground truth marker. Precision and recall versus distance from the marker and the viewing angle are shown in Figure 6.

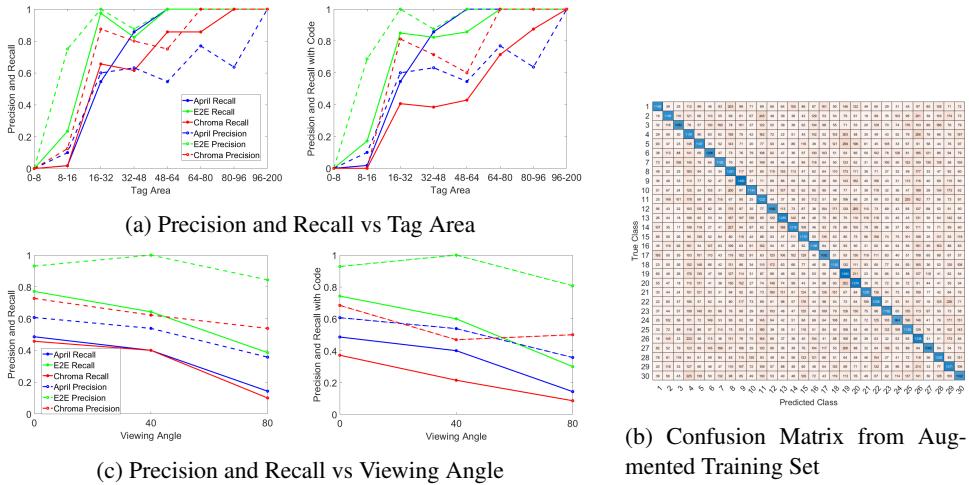


Figure 6: Precision and Recall curves ((a) and (c)) across distance from the marker in meters and viewing angle of the marker in degrees. For each graph AprilTag is blue, ChromaTag is red, and E2ETag is green. Precision and Recall are indicated by a dotted and solid line respectively. The confusion matrix for marker classification is also given in (b).

Classification accuracy is also evaluated using precision and recall. In this case, only markers with the correct class prediction are counted toward the true positive rate. Thus, a false positive has either less than 50% IoU, incorrect class prediction, or both. The results on the left side of Figure 6 do not consider classification errors, while the results on the right side of Figure 6 require correct classification.

Under all conditions, E2ETag has higher precision than AprilTag or ChromaTag. E2ETag also has a higher recall at each viewing angle and for very small tags. AprilTag has higher recall for midrange tag sizes, however, those appear to come at the expense of precision.

While bit encoding methods like AprilTag and ChromaTag are designed to be class-agnostic, E2ETag only encourages this through randomized training. Therefore, to demonstrate the performance of different markers, a confusion matrix (Figure 6b) was generated for all 30 classes. This test is done with Imagenet and COCO background images to adequately gather a large sample size of every marker class. The test is designed to demonstrate performance for difficult predictions. The maximum marker size for this experiment is 32 pixels to encourage a high rate of misclassification. The accuracy ranges from 26.78% to 40.67% across all the markers and the misclassification rate between any two markers ranges from 0.47% to 7.19%. Thus, some markers are more easily classified in highly challenging environments. However, it should be noted that misclassification is rare in practice and often, when detected, the marker is also classified correctly, as illustrated in Figure 6a and 6c.

The results of average precision and recall under various augmentations are shown in Table 1. This table presents results on the original images, as well as those images with isolated augmentations of each type applied to every image in the test set. This includes varying blur, noise, contrast, and white balance.

The most significant detriments to AprilTag are additive noise, large blur, and heavy green white balance. ChromaTag is sensitive to noise and low contrast. In contrast, E2ETag experiences a minimal loss of performance across all augmentations, with the exception of large motion blur. Interestingly, E2ETag has better recall under additive noise, dark contrast,

	Raw	Blur			Noise		Contrast (B,W)				White Balance		
		$l = 5$	$l = 10$	$l = 15$	(-0.15,0.15)	(0.4,1.4)	(0.4,0.6)	(-0.4,0.6)	(-0.4,1.4)	R	G	B	
<b>April</b>	Precision	0.5093	0.5789	0.5303	0.4375	0.4775	0.6047	0.5437	0.8000	0.6477	0.6429	0.4655	0.5392
	Recall	0.3143	0.3143	0.2000	0.1200	0.3029	0.2971	0.3200	0.3886	0.3257	0.3086	0.3086	0.3143
	Precision Code	0.5093	0.5789	0.5303	0.4375	0.4775	0.6047	0.5437	0.8000	0.6477	0.6429	0.4655	0.5392
	Recall Code	0.3143	0.3143	0.2000	0.1200	0.3029	0.2971	0.3200	0.3886	0.3257	0.3086	0.3086	0.3143
<b>Chroma</b>	Precision	0.6375	0.6780	0.6875	0.6944	0.0039	0.6923	0.0000	0.5556	0.2981	0.6000	0.3086	0.7576
	Recall	0.2914	0.2286	0.1886	0.1429	0.1143	0.2571	0.0000	0.2571	0.2743	0.2914	0.2857	0.2857
	Precision Code	0.5397	0.6545	0.6667	0.4211	0.0000	0.4595	0.0000	0.4462	0.2313	0.5526	0.2000	0.5000
	Recall Code	0.1943	0.2057	0.1714	0.0457	0.0000	0.0971	0.0000	0.1657	0.1943	0.2400	0.1600	0.0914
<b>E2ETag</b>	Precision	0.9340	0.9394	0.9302	0.9583	0.9439	0.9412	0.9691	0.9528	0.9196	0.9340	0.9252	0.9333
	Recall	0.5657	0.5314	0.4571	0.3943	0.5771	0.5486	0.5371	0.5771	0.5686	0.5657	0.5657	0.5600
	Precision Code	0.9271	0.9294	0.9130	0.9464	0.9368	0.9259	0.9659	0.9412	0.9072	0.9263	0.9158	0.9271
	Recall Code	0.5086	0.4514	0.3600	0.3029	0.5086	0.4286	0.4857	0.4571	0.5029	0.5029	0.4971	0.5086

Table 1: Results on real images with and without augmentations. Precision Code and Recall Code require correct classification. Each of the augmentations are applied individually to the entire set of Real Images. Specifications are given for the augmentations, where blur is applied at random angles for each length and white balance augments each color channel using R:(1.3,0.7,0.7), G:(0.7,1.3,0.7), and B:(0.7,0.7,1.3).

and high contrast, than on the original images. However, it does have worse classification performance under all augmentations.

To illustrate two ideal success cases, the network outputs are visualized in Figure 7a. The  $80 \times 80$  detection and classification outputs are upsampled and overlaid on the image. Two additional cases for difficult detections under challenging presentations where the method succeeds are shown Figure 7b. In both of these challenging cases, AprilTag and ChromaTag fail to detect the markers. The images in Figure 7c show examples of failed detections where the marker localization does not exceed the required threshold of 0.5. These failures are likely due to the relatively small size of the tags in the image, which are contained within a 12x12 and 18x18 pixel area.

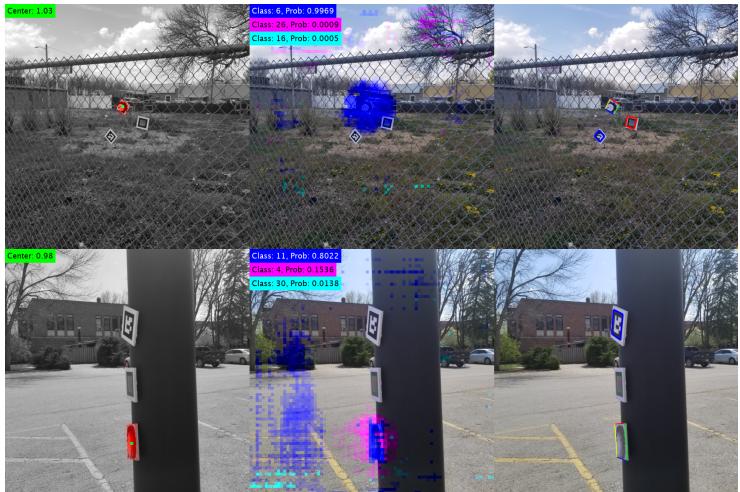
### 4.3 Hardware and Processing Time

The method was implemented with MATLAB using the Deep Learning Toolbox. The computer used for training and forward inference has an Intel i9-9900K 8-core CPU, NVIDIA RTX2080ti GPU, and 16 GB of DDR4 RAM. Detection on a  $640 \times 640$  frame operates at 10 frames per second. ChromaTag and AprilTag, both explicitly designed for efficiency, are considerably faster at 900 fps, and 50 fps, respectively.

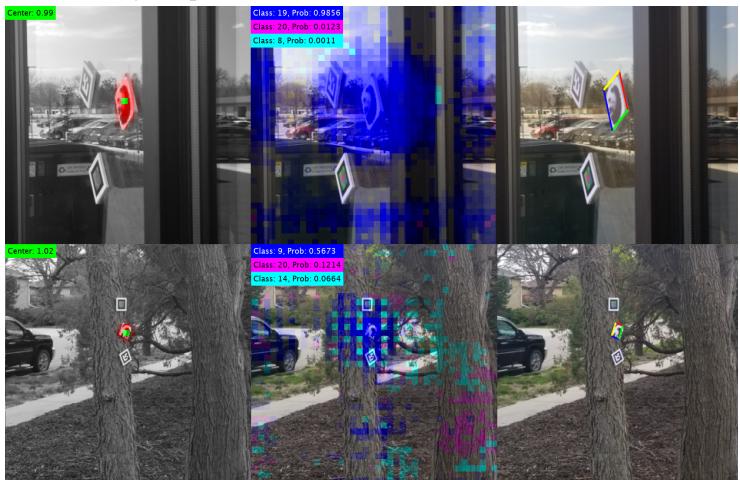
## 5 Conclusion

By training both the marker designs and the detector together under challenging conditions, the method proposed method is able to outperform existing methods at detection and classification. The improvements are especially pronounced for challenging scenes and when images are corrupted by poor exposure, motion blur, and noise. It is noteworthy that the detector was never trained on real images, where the tag was physically placed in the scene, and it is likely that performance would improve even further if the tag designs were fixed and the detector was fine-tuned on real images.

The method is flexible and allows for a wide range of modifications. For example, it would be trivial to change the number of distinct markers, the shape of the markers, or even to place fixed designs into the markers. More generally, the proposed method provides a framework for using deep neural networks to design objects that can be placed and easily detected in the real world.



(a) Easy samples with successful localization and classification.



(b) Difficult samples with successful localization and classification shown at 200% crop.



(c) Samples with failed detections shown at 400% crop.

Figure 7: Leftmost images in each row illustrate the detection channel (red with peak shown in green). The center images of each row illustrate the three largest classification channel output predictions (blue, magenta, cyan). The rightmost images of each row illustrate the projective transformation as colored lines around the marker and AprilTag (blue lines) and ChromaTag (red lines) detections are shown, when detection was successful.

## References

- [1] Bradley Atcheson, Felix Heide, and Wolfgang Heidrich. Caltag: High precision fiducial markers for camera calibration. In *VMV*, volume 10, pages 41–48. Citeseer, 2010.
- [2] Ross Bencina, Martin Kaltenbrunner, and Sergi Jordà. Improved topological fiducial tracking in the reactivision system. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops*, pages 99–99. IEEE, 2005.
- [3] Filippo Bergamasco, Andrea Albarelli, Emanuele Rodola, and Andrea Torsello. Rune-tag: A high accuracy fiducial marker with strong occlusion resilience. In *CVPR 2011*, pages 113–120. IEEE, 2011.
- [4] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *Journal on Image and Video Processing*, 2008:1, 2008.
- [5] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7291–7299, 2017.
- [6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.
- [7] Joseph DeGol, Timothy Bretl, and Derek Hoiem. Chromatag: a colored marker and fast detection algorithm. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1472–1481, 2017.
- [8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [9] Mark Fiala. Artag, a fiducial marker system using digital techniques. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 590–596. IEEE, 2005.
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [11] Martin Kaltenbrunner and Ross Bencina. reactivision: a computer-vision framework for table-based tangible interaction. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 69–74. ACM, 2007.
- [12] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*, pages 85–94. IEEE, 1999.
- [13] Clemens Nylandsted Klokmose, Janus Bager Kristensen, Rolf Bagge, and Kim Halskov. Bullseye: high-precision fiducial tracking for table-based tangible interaction. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*, pages 269–278, 2014.
- [14] Matej Kristan, Jiri Matas, Ales Leonardis, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kamarainen, Luka Čehovin Zajc, Ondrej Drbohlav, Alan Lukezic, Amanda Berg, Abdelrahman Elde索key, Jani Kapyla, and Gustavo Fernandez. The seventh visual object tracking vot2019 challenge results, 2019.

- [15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [16] Yue Liu, Ju Yang, and Mingjun Liu. Recognition of qr code with mobile phones. In *2008 Chinese control and decision conference*, pages 203–206. IEEE, 2008.
- [17] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407. IEEE, 2011.
- [18] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.
- [19] John Wang and Edwin Olson. Apriltag 2: Efficient and robust fiducial detection. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4193–4198. IEEE, 2016.
- [20] Anqi Xu and Gregory Dudek. Fourier tag: A smoothly degradable fiducial marker system with configurable payload capacity. In *2011 Canadian Conference on Computer and Robot Vision*, pages 40–47. IEEE, 2011.