Visual Detection of Doors

Logan Courtney, Ramavarapu Sreenivas

Abstract—Detecting doors in single images can provide useful semantic and geometric information about the scene. Doors are typically non-descript and it is difficult to create a generalized model. Many techniques for detecting doors in the past typically require additional sensors beyond a camera or make a large quantity of assumptions where the usefulness is now limited to the particular domain. This paper investigates methods for automatically detecting a door and compares training various classification techniques. The end result has a successful door detection rate of 93% and false positive rate of 22%. Of the correctly labeled doors, 94% of the door pixels are labeled correctly with only 6% labeled incorrectly.

I. INTRODUCTION AND BACKGROUND

This paper presents a method for detecting doors in indoor environments using only a single image. Knowing the location of doors in the environment and within the image can provide valuable information. Most research combining aspects of robotics and computer vision focus on Simultaneous Localization and Mapping (SLAM) which focuses purely on the 3D geometry of the environment. These methods also typically rely on stereo vision or depth sensors for the added sensing ability with which reliable performance can not be guaranteed. With lack of direct 3D sensing capabilities from a single image, semantic information like door locations could improve navigation and decision making abilities for agents relying on only visual information to complete their assigned task. Man-made environments with doors typically have geometrically consistent properties. Knowing the door locations could provide useful cues for fitting parallel planes to the structure of the room or information on vanishing point locations using the door edges which assist in gathering 3D information from a single image.

Door detection has been studied previously with a wide variety of methods and useful results. In [1], a fuzzy logic approach was used by setting thresholds on various features to determine door labels. It had a 90.5% detection rate for images with doors and 3.5% false detection rate in images with no doors. It was carried out in real time at 6fps. The lack of mathematical model and hand tuned thresholds prevent the method from guaranteeing of success with testing on very large datasets. In [2], an AdaBoost learning algorithm was used to combine the output of a variety of weak classifiers. Features such as the door frame, horizontal lines, door knob, varying color of door/wall, vertical lines, door hinge, low texture on door, door concavity, and door gap were used. The method had only a detection rate of 72% but a very impressive 0.8% false positive rate. However, this approach also utilizes information from a 2D laser range finder and odometer on a mobile robot. In [3], a camera system is used to assist blind persons in navigating by recognizing doors. The method had a 91.9% detection rate with a 2.9% false positive rate. The method hand tunes thresholds using a geometric and visual model of the door. In [4], visual shape and appearance features are used to fit a probabilistic model to a door. Potential door hypotheses are automatically generated and a Gaussian Mixture Model is used to compare histograms of the $L*a*b*$ color space. It has a high detection rate of over 90% but only 79% of the correct pixels are identified and 40% of non-door pixels are contained within the detections. A variable threshold is used to reduce the likelihood of false positives but still has 33% false positives in images with no doors.

This paper pulls various techniques from the research above in an attempt to improve on performance and provide an accurate probabilistic model. The techniques used here are most similar to [4]. A door is represented by four corners and hypotheses are generated to detect all likely combinations of corners within the image. Various features are measured for each hypothesis which are used to evaluate the likelihood of the hypothesis containing a door. The goal is to create a robust, generalized model of a door that allows a classifier to be trained to successfully reject false positives while maintaining a high detection rate. An attempt to reduce the hand-tuning of important thresholds remains of key importance.

In order to create a generalized model, certain features must be avoided. Many techniques for detecting doors with mobile robots use the assumption of being in an indoor corridor where the vanishing point is easily detectable providing some information about the depth or alignment of lines within the image. All the images may be taken from a known height and perspective providing additional features that don’t exist as a standalone feature of a door if viewed from other common views. Features such as the location or absolute size of the door may be useful for most scenarios but they provide no guarantee as the location of a door in an image has nothing to do with what a door actually is, only that it commonly appears in those locations within the image test set.

The paper begins with a probabilistic formulation in section II. The feature selection and hypothesis generation of potential doors using corner points is explained in section III. This is followed with estimating the likelihood functions for typical doors and non-doors within the hypothesis space. Successful results are shown in section IV. A short segment comparing the discriminative approach of neural networks with the generative probabilistic approach is seen in section V. Lastly, it finishes up in section VI discussing particular strengths and weakness of the approach and where to go
from here.

The training data consists of a new dataset of 127 images with 192 doors. The images were taken with a Galaxy S5 in various buildings at the University of Illinois.

II. PROBLEM FORMULATION

For each image, assume we are given or we find a set of hypotheses $H_d$ consisting of subsets of pixels contained within the image. These subsets of pixels are bounded by four corners. Each hypothesis $h \in H_d$ is to be classified as either a door or non-door. A hypothesis consists of measurements $X = \{x_1, x_2, \ldots, x_n\}$ which are chosen features representing our model of a door. We are interested in calculating the posterior probability $P(D|X)$ of the hypothesis being a door given the measurements.

$$P(D|X) = \frac{P(D)P(X|D)}{P(X)}$$

The evidence distribution $P(X)$ can be found using the prior distribution and likelihood.

$$P(X) = P(D)P(X|D) + P(\overline{D})P(X|\overline{D})$$

The features are assumed to be independent of one another splitting the likelihood function into multiple parts.

$$P(X|D) = P(x_1, x_2, \ldots, x_n|D)$$

$$= P(x_1|D)P(x_2|D, x_1) \ldots P(x_n|D, x_1, x_2, \ldots, x_{n-1})$$

$$= P(x_1|D)P(x_2|D) \ldots P(x_n|D)$$

$$= \prod_{i=1}^{n} P(x_i|D)$$

At this point, the posterior probability can be rewritten in terms of the likelihood ratio $LR(X|D)$ except that it is normalized by the evidence making it a valid distribution.

$$P(D|X) = \frac{P(D)P(X|D)}{P(D)P(X|D) + P(\overline{D})P(X|\overline{D})}$$

$$= \frac{LR(D)LR(X|D)}{LR(D)LR(X|D) + 1}$$

$$LR(X|D) = \frac{P(X|D)}{P(X|\overline{D})}$$

$$LR(D) = \frac{P(D)}{P(\overline{D})}$$

A. Maximum Likelihood Estimation (ML)

Our first approach assumes equal prior distributions $P(D) = P(\overline{D})$ simplifying the posterior probability.

$$P(D|X) = \frac{LR(X|D)}{LR(X|D) + 1}$$

We are left with the following decision rule.

$$d_{ML} = \begin{cases} 1 & P(D|X) \geq 0.5 \text{ or } LR(X|D) \geq 1 \\ -1 & P(D|X) < 0.5 \text{ or } LR(X|D) < 1 \end{cases}$$

For convenience, we will use the log-likelihood ratio

$$LLR(X|D) = \ln(LR(X|D)) = \sum_{i=1}^{n} LLR(x_i|D)$$

Note the above technique provides the same results as if it was formulating using $d_{ML} = \text{argmin}\{\ln(P(X|D))\}$. However, the normalization of the posterior probability allows for a comparison between the likelihood ratio of different hypotheses within the same image. The hypotheses are technically not independent. It is possible for the pixels belonging to one hypothesis to be a subset of the pixels belonging to another hypothesis. If these pixels happen to belong to a door, its reasonable to assume the maximum likelihood classification would label both hypotheses as a door. In this case, the likelihood ratios can be compared to throw out the less likely of the two.

B. Maximum a Posteriori Estimation (MAP)

Later in section III-D.5 we will see there is an imbalance in the prior distributions that can be found which has a significant effect on the performance of the classifier. Once it is known, the decision rule can be changed accordingly.

$$d_{MAP} = \begin{cases} 1 & LLR(D) + \sum_{i=1}^{n} LLR(x_i|D) \geq 0 \\ -1 & LLR(D) + \sum_{i=1}^{n} LLR(x_i|D) < 0 \end{cases}$$

This is essentially only changing the threshold for positive and negative classification.

$$d_{MAP} = \begin{cases} 1 & \sum_{i=1}^{n} LLR(x_i|D) \geq -LLR(D) \\ -1 & \sum_{i=1}^{n} LLR(x_i|D) < -LLR(D) \end{cases}$$

III. APPROACH

A. Feature Selection

Four features were used to determine the likelihood of a door: aspect ratio $(\text{width/height}) X_{AR}$, gradient magnitude uniformity $X_{GM}$, pixel distance from the closest edge $X_{ED}$, and pixel luma $X_{PL}$. Although the color of doors can be very descriptive, they were avoided due to a small set of only 127 training images and 191 doors. With such a small dataset, it is unlikely the "true" color of doors can be learned. If the dataset is not a general enough representation, the classifier can become dependent on the color by dominating the other features and yielding highly inaccurate results when encountered with underrepresented door colors. The features chosen are an attempt to provide a general yet effective door model when paired with the appropriate hypothesis generating technique.
1) Aspect Ratio: A rectangle in 3D space will project to a quadrilateral in the image plane. A single camera does not provide enough information to determine the width and height separately. However, given the location of the four corners in the image plane, the aspect ratio \((w/h)\) can be determined under the assumption that the four corners actually represent a rectangle in 3D space.

Fig. 1. Diagram representing the perspective projection of a rectangle in 3D space onto an image plane.

Let \(\vec{M} = \{(0, 0), (w, 0), (0, h), (w, h)\}\) represent the \((x, y)\) locations of the door corners in 3D space all at a depth of \(z = 0\). Let \(\vec{m} = \{(x_1, y_1, 1), (x_2, y_2, 1), (x_3, y_3, 1), (x_4, y_4, 1)\}\) represent the augmented vectors with an appended third coordinate equal to 1 representing the location of the four corners in the image plane. The projection from \(\vec{M}\) to \(\vec{m}\) can be represented by the following model

\[
R = \begin{bmatrix} r_1 & r_2 & r_3 \\
\end{bmatrix}
\]

\[
\lambda\vec{m} = A[R t] \vec{M}
\]

where \(R\) and \(t\) are the rotation matrix and translation vector from the world coordinate frame to the image coordinate frame and \(A\) is the calibration matrix of the camera.

With a known calibration matrix and image coordinates, the aspect ratio can be determined.

\[
\frac{w^2}{h^2} = \frac{n_2'^2A^TA^{-1}n_2}{n_3'^2A^TA^{-1}n_3}
\]

\[
n_2 = k_2\vec{m}_2 - \vec{m}_1
\]

\[
n_3 = k_3\vec{m}_3 - \vec{m}_1
\]

\[
k_2 = \frac{(\vec{m}_1 \times \vec{m}_4) \cdot \vec{m}_2}{(\vec{m}_2 \times \vec{m}_4) \cdot \vec{m}_3}
\]

\[
k_3 = \frac{(\vec{m}_1 \times \vec{m}_4) \cdot \vec{m}_2}{(\vec{m}_3 \times \vec{m}_4) \cdot \vec{m}_2}
\]

The details can be found [5].

2) Gradient Magnitude Uniformity: Although absolute models of door color are being avoided, the relative change of pixel intensity within a single image provides valuable information.

Fig. 2. The gradient of the input image on the left is shown on the right with white pixels representing a large change in pixel intensity.

The bottom segment of a door typically has very little texture and avoids noticeable changes caused from the door handle or potential windows in the upper door segment. The gradient of an image shows relative change in a small neighborhood around each pixel. The gradient can be relatively large in one image compared to another. Therefore, the variance of the gradient is calculated representing the deviation only within the red area which remains relatively similar between images. This feature helps eliminate large hypotheses that overlap edges.

Fig. 3. The variance of the gradient within the red box is used as a feature in the door model.

3) Pixel Distance from Closest Edge: Suitable corners within an image will be shown to generate hypotheses in the later section. Corners of doors distinguish themselves typically by falling on the edge boundary that is usually apparent between the door and wall. A canny edge detector uses the gradient of an image to create a binary mask of ones and zeros with ones signifying pixels likely to be on an edge. The distance from each zero pixel to edge pixel is calculated and stored within a matrix of the same size as the image. Each hypothesis consists of four corners as well as the four lines connecting these corners. The average distance from each pixel within a line to the closest edge pixel is used as the third feature.

4) Pixel Luma: The input RGB image can be converted to the \(L^*a^*b^*\) color space which contains all the colors
visible by the human eye and separates chrominance from luminance. The $L^*$ channel is meant to represent only the light intensity and no color information. In indoor environments, the lights are typically pointing downward meaning the light intensity of the floor and ceilings is relatively high compared to the rest of the scene. Additionally, additional shading is caused by the door frame from the door being built into the wall. A histogram was found from the training images of all of the door pixels and nondoor pixels which is used for the fourth feature with further details being explained in the later sections.

B. Hypothesis Generation

A door in the image will be represented by its four corners. Individual corners are found and matched into sets of four making up the hypothesis set for each image. The naive approach would be to find every hypothesis possible given the $n$ corners detected in the image. This is on the order of $n^4$ which is quite costly considering the number of corners in an image can be in the hundreds. Instead, a series of techniques are used to classify the corners into different types and only combine corners that form reasonable estimates of door locations. The hypothesis generation proves to be extremely sensitive to the choice of parameters while having a significant impact on the performance of the classifier. The same set of parameters for the hypothesis generation is used for every image tested.

1) Edge Detection and Line Fitting: A binary mask from the canny edge detection is used in a line segment fitting algorithm discussed in [6] which proved more reliable than a standard hough lines transform which often can have spurious line segments and/or broken line segments. The segments are split into vertical and horizontal lines based on their angle in the image frame. The canny edge threshold is set low enough to not miss edges in low light conditions which means occasionally the edge map picks up on highly textured floors or objects. Because of this, only vertical lines greater than 60 pixels ($\frac{1}{16}$ of the image height) and horizontal lines greater than 32 pixels ($\frac{1}{20}$ of image width) are kept.

2) Corner Detection and Classification: The intersection point (or predicted intersection if the lines are not touching) between every vertical and horizontal line is considered a potential corner. If the center point of the horizontal line is above the center point of the vertical line, the corner is classified as a top ($T$) corner while it is classified as a bottom ($B$) otherwise.

There remains a lot of unnecessary corners before any sort of filtering as can be seen in the left window of figure 7. The Harris Corner Measure is matrix of values signifying large changes of the gradient in multiple directions at each pixel. Corners having a value of $<0.01%$ of the maximum value in the Harris Corner Measure matrix are removed. Additionally, intersection points are removed if its location is a significant distance away from the lines that caused it. A short horizontal segment on one side of the image should not create a potential corner with every vertical segment in the image. Lastly, if the intersection point does not possess the maximum value from the Harris Corner Measure in a small neighborhood (19x19 matrix in this case) around itself, it is removed. The remaining corners are used for generating
hypotheses with an example being shown in the right window of figure 7.

It may seem unnecessary to generate corners this way instead of just using the endpoints of the line segments or modeling the doors with the lines themselves instead of corner points. However, as mentioned previously, the edge detection and line fitting are sensitive to the choice in parameters from image to image. The lines forming the door are often only partial segments or missed entirely. By using intersection points instead of lines or the endpoints of lines, it is still possible to capture the full door in the generated hypothesis space without which the classifier would never have the chance to detect. Figure 8 illustrates an example of this.

Fig. 8. The above image illustrates a missing door boundary due to the color of the door being nearly the same as the wall. Using the predicted intersection points of the lines as our door model instead of the lines allows the full door to remain detectable.

3) Corner Combinations: Using reasonable heuristics for combining corners into hypotheses creates a trade off between reduction in computation time and restrictions on the doors capable of being detected. All suitable combinations of top corners \( t \in T \) and bottom corners \( b \in B \) are combined to form a set \( H_2 = \{(t, b)\} \). In this case, suitable means the distance between the two corners is greater than 120 pixels (\( \frac{1}{6} \) of the image height) and the angle formed by the two corners is within 8\(^\circ\) of the vertical lines forming the two corners. This drastically reduces computation time while creating only a slight restriction that doors with a height smaller than \( \frac{1}{3} \) of the image cannot be detected. Figure 9 shows an example of the top-bottom pairs. There are 70 pairs in the right image based on the 45 top corners and 41 bottom corners instead of the 1845 possible combinations in the left image (which would turn into millions of four corner hypotheses before evaluating the likelihoods).

Fig. 9. The left image shows all possible combinations of top and bottom corners. The right image shows the remaining combinations of top and bottom corners after removing unlikely pairs.

The three corner set \( H_3 \) consists of every member of \( H_2 \cap (T \cup B) \) satisfying additional constraints. The new horizontal line formed by the top-top pair or bottom-bottom pair has line length \( l \) such that \( 20 < l < 300 \) and the angle formed by the two corners must be within 20\(^\circ\) of the horizontal lines forming the two corners. These are loose constraints having little impact on the doors capable of being detected but still greatly reduces the amount of feasible hypotheses. Figure 10 shows an example of the set \( H_3 \).

Fig. 10. The image shows the three corner set \( H_3 \). The yellow lines show the two corner set \( H_2 \) while the blue lines represent the newly added corner whether it be a top corner (red) or bottom corner (blue).

The final four corner hypothesis set \( H_4 \) is formed from every member of \( H_3 \cap (T \cup B) \) satisfying the same constraints used for forming the three corner set. Due to the fourth corner of a door occasionally being occluded by another object or simply not being in the image, \( H_4 \) also includes the hypotheses formed by predicting where the last corner would be. For example, if a member of \( H_3 \) consists of two top corners and one bottom corner, the intersection point between the horizontal line that caused the bottom corner with the vertical line that caused the open top corner is found and assumed to complete the hypothesis. Figure 11 shows the set \( H_4 \). Note the newly formed blue lines extending out of the screen for the occluded bottom corner of the left-most door.

Fig. 11. The image shows the four corner set \( H_4 \). The yellow lines show the three corner set \( H_3 \) while the blue lines represent the newly added corner. It is possible for the set \( H_4 \) to have one occluded or missing corner.

C. Generating Training Data for Likelihood Estimation

The goal is to somehow find a way to estimate the likelihood functions \( P(X|D) \) and \( P(X|\overline{D}) \) that allow for the evaluation of every hypothesis \( h \in H_4 \). With the positive labels from the training images, it would be possible to find \( P(X|D) \). However, it is unlikely the hypothesis generation will form the exact hypothesis that matches the
training image. Fitting too tight of a distribution to \( P(X|D) \) could cause “good enough” hypotheses to be considered a non-door label. Additionally, generating negative labels for determining \( P(X|\overline{D}) \) is troublesome. Randomly generated corners in each image used for negative labeled data could cause fitting too loose of a distribution to \( P(X|\overline{D}) \) since the hypothesis generation is not random. This could cause a large amount of false positives since the hypothesis set contains far better door candidates than a random set would.

The most reasonable option is to label every member of the generated hypothesis set \( H_4 \) as positive or negative to be used for finding \( P(X|D) \) and \( P(X|\overline{D}) \). Hand labeling is infeasible considering there are typically 100 to 1000 hypotheses per image. In order to use automatic labeling of the hypotheses from the door labeled training set, a suitable definition for what is and is not considered a door is necessary. If a hypothesis covers an entire image which includes a door, what should it be labeled? If a hypothesis covers a partial door, what should it be labeled?

This choice has a significant impact on the strength of each feature and the overall performance of the classifier. If all hypotheses with greater than 50\% door pixels are considered positive, the pixel luma feature \( X_P \) would be very important while it would weaken features like the aspect ratio \( X_A \) and edge pixel distance \( X_E \). The aspect ratio of partial doors is not typical of an actual door and partial doors typically do not fall on edges.

1) **Automatic Hypothesis Labeling for Training Set**: Let \( S_d \) represent the number of pixels in a single labeled door \( d \) within a training image. Let \( S_h \) represent the number of pixels covered by a hypothesis \( h \in H_4 \). Let \( n_d \) be the number of door pixels contained within the hypothesis \( h \). Let \( n_{\overline{D}} \) represent the number of non-door pixels contained with the hypothesis \( h \).

\[
\begin{align*}
  d_h &= \frac{n_d}{S_d} \\
  \overline{d}_h &= \frac{n_{\overline{D}}}{S_h}
\end{align*}
\]

Note the percentage of door pixels \( d_h \) is calculated with respect to the number of pixels in the labeled door \( S_d \) while the percentage of non-door pixels \( \overline{d}_h \) is calculated with respect to the number of pixels in the hypothesis \( S_h \). Thresholds are used on these two values to automatically label the generated hypotheses as positive or negative. Let \( d_h \) be the automatic label from the training data follow the decision rule below.

\[
d_h = \begin{cases} 
  1 & d_h \geq 0.80 \text{ and } \overline{d}_h < 0.20 \\
  -1 & d_h < 0.80 \text{ or } \overline{d}_h \geq 0.20 
\end{cases}
\]

Essentially, the first stage of the door detection is automatically generating the hypothesis set \( H \) with two ideas in consideration. There should be a very high true detection rate where the probability of there existing a hypothesis containing nearly every door pixel and relatively few non-door pixels is maximized while the total amount of hypotheses is minimized to reduce computation time. This means there will be a large amount of incorrect hypotheses in order to ensure at least one true hypothesis exists. This is done by the methods laid out in section III-B.

The second stage is automatically creating the training labels \( d_h \) which should create a training set with very few false positives while not worrying as much about missed detections. The total hypothesis set from the first stage is primarily non-door hypotheses, meaning missed detections would have relatively impact when training the classifier while false positives would greatly weaken the classifier since there are fewer positive hypotheses to train from. This is done by choosing the thresholds for \( d_h \) and \( \overline{d}_h \).

The third stage is training the classifier based on the automatically labeled hypothesis data. The likelihood \( P(X|D) \) will match closely to the true likelihood function of hypotheses being doors considering there are few false positives while \( P(X|\overline{D}) \) will match closely to the true likelihood function for hypotheses being non-doors considering true negatives are far more frequent than missed detections.

\[
\text{Fig. 12. The image shows another representation of the four corner set } H_4 \text{ where the green intensity represents the overlapping hypotheses.}
\]

\[
\text{Fig. 13. The figure shows } d_h \text{ on the horizontal axis with } \overline{d}_h \text{ on the vertical axis for every generated hypothesis from the training images. The red points signify } +1 \text{ for the automatically generated training label } d_h \text{ while the blue points signify } -1 \text{ for the label.}
\]

With these thresholds set for \( d_h \) and \( \overline{d}_h \), of the doors in the training images have at least one hypothesis labeled as true. The 10 missing doors are double doors without a detectable edge between the two to generate the hypothesis like the one shown in figure [14]. These doors typically have
a hypothesis covering both of them causing \(d_g \approx 100\) and \(d_g \approx 50\) meaning it will be labeled as a non-door before training.

Fig. 14. The figure shows the detected lines and corners points of an image with two individually labeled doors. The edge between them is indistinguishable meaning no hypothesis is generated capable of individually recognizing the doors.

D. Likelihood Ratio Estimation

It is possible to fit either a Gaussian distribution or an exponential distribution to the likelihood \(P(X|D)\) for the positive labels. However, the likelihood \(P(X|\overline{D})\) for the negative labels does not have as smooth of data to work with. Additionally, fitting unimodal distributions for the likelihood in either case caused too quick of a drop-off at the tails of the distributions to provide useful results. With a larger amount of training data, it’s most likely each distribution could be well represented as a mixture of Gaussian distributions. We do not have enough training data to capture this information and instead there are small anomalies showing up as outliers. These show up in the tails of the unimodal distributions for \(P(X|D)\) which implies they are highly unlikely to be doors which is not actually the case. Because of this, it was much easier to simply create a distribution from a histogram of the different features that accurately represented the data from underrepresented modes in the true distribution.

Fig. 15. \(P(X_{AR}|D)\) and \(P(X_{AR}|\overline{D})\) are on the left and the likelihood ratio \(LR(X_{AR}|D)\) is shown on the right.

1) Aspect Ratio Likelihood: The mean of the aspect ratio for positive labeled data falls between 0.4 and 0.5. This follows the intuition for picking the feature in the first place of doors typically being 2 to 3 times as tall as they are wide.

2) Gradient Magnitude Likelihood: The mean of the gradient for the bottom section of the hypotheses was typically a lot smaller for the positive labeled data. The bottom half of doors typically has very little texture.

Fig. 16. \(P(X_{GM}|D)\) and \(P(X_{GM}|\overline{D})\) are on the left and the likelihood ratio \(LR(X_{GM}|D)\) is shown on the right.

3) Pixel Distance from Closest Edge Likelihood: The mean distance from the closest edge pixel is smaller for the positively labeled data but not quite as powerful of a feature as expected. This is most likely due to the fact that the hypotheses are generated using the edges in the first place meaning most of them are close. It still is distinctive enough to assist the classification model.

Fig. 17. \(P(X_{ED}|D)\) and \(P(X_{ED}|\overline{D})\) are on the left and the likelihood ratio \(LR(X_{ED}|D)\) is shown on the right.

4) Pixel Luma Likelihood: Previously, figure [5] showed the histogram distribution of light intensity for door pixels \((f_D)\) and non-door pixels \((f_{\overline{D}})\). Each individual hypothesis also has two light intensity distributions, one for pixels contained within the hypothesis \((f_{dh})\) and one for the remaining pixels outside of the hypothesis \((f_{\overline{dh}})\). These distributions are compared by taking the Bhattacharyya distance \(BD(f_1, f_2)\) and then combined to form the feature \(X_{PL}\).

\[
BD(f_1, f_2) = 1 - \sum_{i=1}^{n} \sqrt{f_1(i)f_2(i)}
\]

The index \(i\) is the number of bins and \(f(i)\) is the probability of the \(i^{th}\) bin. A Bhattacharyya distance of 1 implies the distributions are completely different while a Bhattacharyya distance of 0 implies the distributions are the same.
• $BD(f_{dh}, f_D)$: Represents difference between hypothesis door pixels and general distribution of door pixels (should be low for correct door hypotheses)
• $BD(f_{dh}, f_{\overline{D}})$: Represents difference between hypothesis door pixels and general distribution of non-door pixels (should be high for correct door hypotheses)
• $BD(f_{\overline{dh}}, f_D)$: Represents difference between hypothesis non-door pixels and general distribution of door pixels (should be high for correct doors)
• $BD(f_{\overline{dh}}, f_{\overline{D}})$: Represents difference between hypothesis non-door pixels and general distribution of non-door pixels (should be low for correct doors)

$$X_{PL}(h) = (BD(f_{dh}, f_D) + BD(f_{\overline{dh}}, f_{\overline{D}})) - (BD(f_{dh}, f_{\overline{D}}) + BD(f_{\overline{dh}}, f_D))$$

Figure 19 shows the likelihood functions as well as the likelihood ratio for this feature.

![Figure 19](image1.png)

Fig. 19. $P(X_{PL}|D)$ and $P(X_{PL}|\overline{D})$ are on the left and the likelihood ratio $LR(X_{PL}|D)$ is shown on the right.

5) Prior Distribution: For the 127 training images, there were a total of 59,748 hypotheses for an average of $\approx 500$ hypotheses per image. The amount of negative labels greatly outnumber the positive labels allowing the prior distributions to be estimated as $P(D) = 0.1141$ and $P(\overline{D}) = 0.8859$. This has no effect on the ML decision rule but alters the MAP decision rule.

$$LR(D) = \frac{P(D)}{P(D) + P(\overline{D})} \implies LLR(D) = -2.05$$

$$d_{MAP} = \begin{cases} 1, & \sum_{i=1}^{n} LLR(x_i|D) \geq 2.05 \\ -1, & \sum_{i=1}^{n} LLR(x_i|D) < 2.05 \end{cases}$$

IV. RESULTS

A. Maximum Likelihood Classification

Figure 20 shows the results of the ML classifier of every hypothesis for training. However, this does not represent the actual performance of the classifier.

Many of the hypotheses are overlapping. When testing the classifier, the most positive hypothesis is labeled as a door. Any overlapping hypothesis at this point is thrown out. This process is continued until no hypotheses with a positive output are remaining.

![Figure 20](image2.png)

Fig. 20. The figure shows the log likelihood output after training. The x axis is $d_{\overline{D}}$, the y axis is $f_{\overline{D}}$, and the z axis is the log likelihood ratio output. Blue points were negative labeled hypotheses and red points were positive labeled hypotheses.

<table>
<thead>
<tr>
<th>Detection Rate</th>
<th>Correct Pixels</th>
<th>False Pixels</th>
<th>False Positive Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>98.44%</td>
<td>93.89%</td>
<td>6.94%</td>
<td>0.9921</td>
</tr>
</tbody>
</table>

The detection rate is the percentage of all doors detected (192 possible in the 127 images). Correct Pixels is the door coverage percentage for only the true detection hypotheses. The false pixels is the non-door coverage percentage for only the true detection hypotheses. The false positive rate is the amount of false detections (regardless of how large the hypothesis is) per image.

B. Maximum a Posteriori Classification

The ML classifier had a significant amount of false positive detections (almost an average of 1 per image). The MAP classifier greatly reduced this amount while only slightly reducing the door detection rate. The accuracy of the correct hypotheses remained unaffected.

<table>
<thead>
<tr>
<th>Detection Rate</th>
<th>Correct Pixels</th>
<th>False Pixels</th>
<th>False Positive Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>93.23%</td>
<td>94.23%</td>
<td>6.45%</td>
<td>0.225</td>
</tr>
</tbody>
</table>

C. Examples

V. NEURAL NETWORK (NN)

Under the assumption the likelihood functions forming $P(X|D)$ are belonging to the exponential family which is presumably the case based on the training data, it is possible to show the maximum likelihood criterion produces the same result as some linear combination of the features passed through a differentiable nonlinear function.

The ML and MAP approach provide some convenience in the sense that a closed form decision rule is found maximizing the probability of correctly labeling a hypothesis. However, as was shown earlier, it is difficult to correctly identify these likelihood distributions accurately. The results were poor when fitting unimodal models to the likelihood functions. The training data in this case may not be representative of the true underlying distributions for typical doors.

Training a neural network classifier eliminates the difficult of fitting distributions while still capable of producing the same result.
Using the four features from before along with one bias feature set to 1, two neural networks with four input nodes and one hidden layer were trained. The first NN contained only a single hidden node making it simply a linear classifier. The second NN contained four hidden nodes to possibly find some classifier that may be a nonlinear function of the inputs.

Training a simple feed-forward neural network with back propagation on an unbalanced dataset will typically cause problems with the output favoring the more represented class. Therefore, a repeated cycle of randomly sampling the negatively labeled hypotheses to match the number of positively labeled hypotheses was used to train the network until the error function became stagnant.

There was essentially no difference between the neural networks and both performed with similar results to the ML approach with a large number of false positives.

<table>
<thead>
<tr>
<th>Num of Hidden Nodes</th>
<th>1</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detection Rate</td>
<td>97.40%</td>
<td>97.40%</td>
</tr>
<tr>
<td>Correct Pixels</td>
<td>95.33%</td>
<td>95.58%</td>
</tr>
<tr>
<td>False Pixels</td>
<td>9.40%</td>
<td>9.61%</td>
</tr>
<tr>
<td>False Positive Rate</td>
<td>0.7323</td>
<td>0.7323</td>
</tr>
</tbody>
</table>

Although the number of false positives was high, most of them had significantly lower outputs than the correctly labeled examples. Thresholds were chosen to maximize the probability of correct labels by weighting false positives equal with missed detections. The results improved significantly. However, without an image test set, it is uncertain whether this threshold is a true parameter or simply over learning the training set.

<table>
<thead>
<tr>
<th>Num of Hidden Nodes</th>
<th>1</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detection Rate</td>
<td>90.63%</td>
<td>89.58%</td>
</tr>
<tr>
<td>Correct Pixels</td>
<td>95.82%</td>
<td>95.96%</td>
</tr>
<tr>
<td>False Pixels</td>
<td>9.44%</td>
<td>9.64%</td>
</tr>
<tr>
<td>False Positive Rate</td>
<td>0.1417</td>
<td>0.1339</td>
</tr>
</tbody>
</table>

Although neural networks can sometimes do too much behind the scenes without providing intuition of the model, it’s effectiveness is demonstrated purely by it’s ability to produce comparable results while requiring little to no work for the implementation and no trouble with attempting to fit distributions to the results.
It is understandable that the network with more hidden nodes had nearly identical performance to the single node network. Three out of four of the features have thresholds where all values below are more likely to represent a door while all values above likely represent a non-door. There most likely is not a nonlinear combination of the features used that would provide more successful results.

VI. DISCUSSION AND CONCLUSION

A method for generating and evaluating hypotheses was discussed with successful results given the training data. More images will need to be gathered to determine the strength of the classifier and whether it generally represents the real life model of a door.

The main weaknesses do not lie within the classification methods. As was shown in figure 23 and figure 24 there are commonly seen objects in indoor environments that are indistinguishable from doors if only looking at the features used. Additionally, as mentioned previously, relying on the color or light intensity of doors has significant drawbacks considering these can be drastically different between environments. Either more images need to be gathered from a wide variety of scenes and the pixel intensity likelihood fit with something like a Gaussian mixture model or additional features need to need to be used to make up for this sensitivity. Features measured relative to individual images are significantly more general and useful than using average data over all the images. For example, instead of finding the distribution of color for all doors, it may be more practical to use a feature where doors simply need to be a different color than its surroundings as this is more robust to doors that have not been seen before in the training set.

Methods for generating non-door hypotheses for training can be improved as well. The number of hypotheses per image is around 500 but this can range from 50 to over a thousand. This means the classifier is possibly presented with far more hypotheses of one setting than others which could definitely skew the results or explain the anomalies in the distribution making them appear multimodal.

REFERENCES


