Towards the Recognition of Sketches by Learning Common Qualitative Representations

Wael Zakaria[1] Ahmed M. H. Abdelfattah[1] Nashwa Abdelghaffar[1] Nermeen Elsaadany[1] Nohayr Abdelmoneim[1] Haythem Ismail[2] Kai-Uwe Kühnberger[3]

Faculty of Science, Ain Shams University, Egypt Faculty of Engineering, Cairo University, Egypt
Institute of Cognitive Science, University of Osnabrück, Germany

Abstract.

The aim of this paper is to take advantage of major perspectives for processing hand-drawn sketches to build a hybrid technique, in which aspects of machine learning, computer vision, and qualitative representations are mixed to produce a classifier for sketches of four specific types of objects.

Keywords: Feature learning - HOG - spatial relations - sketch recognition

1 Introduction and Problem Definition

Sketch recognition is a well-established field in AI and spatial cognition, where techniques that recognize hand-drawn sketches usually follow one of two major perspectives for processing common sketches: raster and vector. The former considers a sketch as an image, from which a feature vector is built from a histograms of oriented gradient (HOG) of pixels. Then, it uses a machine learning technique, such as support vector machines, to build a classifier model. Vector processing, on the other hand, deals with a sketch as a set of points, from which qualitative representations are extracted to build a knowledge base of features, on which analogical processing models can be applied for the recognition process. (cf. §4 for some references to existing work in sketch recognition.)

This paper presents a hybrid technique, QuRSeR-Tech, that is based on qualitative representation for recognizing hand-drawn sketches. The presented technique goes somewhat down into low-level representations of sketches, where a sketch consists of geometric shapes as the sketch's constituents that have simple QR features such as relative sizes of geometric shapes, positional relations of any successive pairs of geometric shapes, and similarity group of any number of successive geometric shapes that have the same relative sizes. Our technique has two SVM-based classifiers that employ histograms of oriented gradient, HOG, characteristics from computer vision, and a restricted set of qualitative representations (QR) [Scholkopf and Smola, 2001,Dalal et al., 2006,Cohn and Renz, 2008].

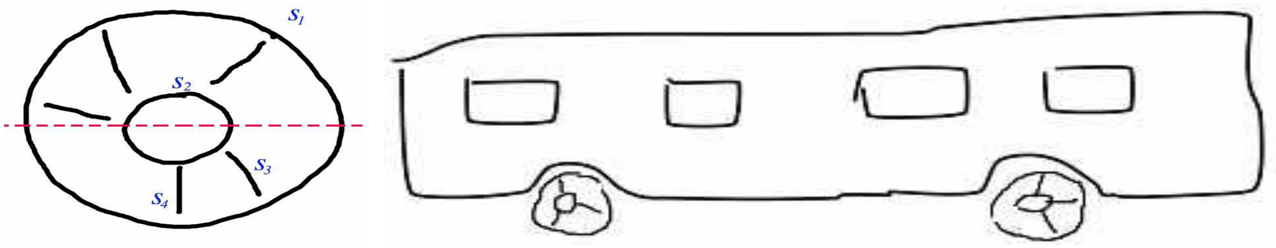
1.1 Abstract Building Blocks

In the following, we formulate the basic definitions of the terms we use in this paper, and explain some conventions that help understanding the rest of the presentation and the obtained results.

A stroke is the basic building block of a sketch. It can be represented as a sequence of points in the sense of a polyline, but we formally define a **stroke** as a sequence $s := \langle p_i \rangle_{i=1}^{n(s)}$ of $n(s) \in \mathbb{N}$ ordered tuples that correspond to sequentially recorded points between a pen-down and a pen-up events. Each representation of a point p_i corresponds to a tuple containing both the x_i- and y_i- coordinates of the pixels composing s, as well

In: A.M. Olteteanu, Z. Falomir (eds.): Proceedings of ProSocrates 2017: Symposium on Problem-solving, Creativity and Spatial Reasoning in Cognitive Systems, at Delmenhorst, Germany, July 2017, published at http://ceur-ws.org

as a temporal parameter, t_i, that reflects the time of drawing each pixel p_i, for $1 \leq i \leq n(s)$. For simplicity, we may assume that a stroke s is a sequence of $n(s)$ ordered pairs, $s := \langle (x_i, y_i) \rangle_{i=1}^{n(s)}$, with $p_1 = (x_1, y_1)$ being the pair corresponding to the coordinates of the "start point" of drawing s (pen down), and $p_{n(s)} = (x_{n(s)}, y_{n(s)})$ corresponding to s's last point (pen up). Even though, p_1 and $p_{n(s)}$, in particular, must also reflect (at least) the temporal parameters, t_1 and $t_{n(s)}$, that record the start and end times, respectively, of s. We also define a **sketch** as a sequence of strokes, $\mathcal{S} = \langle s_1, s_2, \ldots, s_{m(\mathcal{S})} \rangle$, where i is the stroke number for each $1 \leq i \leq m(\mathcal{S})$. Two strokes s_i and s_{i+1} are called *successive*. Figure 1a gives an example of a sketch, $\mathcal{S}_{w_1} = \langle s_1, s_2, s_3, s_4, s_5, s_6, s_7 \rangle$ that contains seven strokes (i.e., $m(\mathcal{S}_{w_1}) = 7$) corresponding to two successive "ellipses" (circles) followed by five consecutive "line segments".



(a) The sketch $\mathcal{S}_{w_1}$

(b) The sketch $\mathcal{S}_{b_1}$

Fig. 1: Two hand-drawn sketches of two different objects: (a) a WHEEL object (which is referred to in the text as $\mathcal{S}_{w_1}$, with some illustrative labels and a dashed line), and (b) a BUS object (referred to as $\mathcal{S}_{b_1}$).

To assist the learner presented later in §2, we base the building on a **feature space** that contains guiding information about all sketches of the set of objects to be learnt (cf. Figure 2). From all the sketched objects, we gather and record a specific number of qualitative representational (QR) criteria (which are limited in this paper to the following three kinds): *relative size*, *positional relation*, and *similarity group*. Relative sizes measure the drawn stokes' metric lengths compared to that of the whole sketch they compose. The size of the (whole) sketch is measured by calculating the perimeter of the canvas that contains it. Based on this size, values ranging from zero to the perimeter's are discretized into intervals, each of which is assigned to one of the **relative sizes**: tiny, small, medium, semi-medium, large, very large, and huge. The relative size of a stroke is assigned to the intervals it belongs to. A **positional relation** is a binary relation between (the centroids of) the two geometric shapes corresponding to successive strokes in a sketch. For a sketch $\mathcal{S} = \langle s_1, s_2, \ldots, s_{m(\mathcal{S})} \rangle$, positional relations $R(s_{i+1}, s_i)$ on $\mathcal{S} \times \mathcal{S}$ determine the location of a stroke s_{i+1}, relative to that of the stroke immediately sketched before it, s_i, for $1 \leq i < m(\mathcal{S})$, where the positional relation R can be one of the following: equal ($\equiv$), up ($\uparrow$), right ($\rightarrow$), left ($\leftarrow$), up-left ($\nwarrow$), up-right ($\nearrow$), down ($\downarrow$), down-left ($\swarrow$), and down-right ($\searrow$). A **similarity group** is a collection of two or more geometric shapes of the same relative sizes. In other words, successive geometric shapes of the same type (e.g., ellipses) can be grouped together into one group if they have the same relative size.

For each sketch $\mathcal{S} = \langle s_1, s_2, \ldots, s_{m(\mathcal{S})} \rangle$, we record a (constant) number M of selected features that reflect the qualitative representation of a sketch $\mathbf{QR}(\mathcal{S})$. A QR feature vector, $f_{\mathcal{S}}$, is used for recording numerical values $\langle f_i \rangle_{i=1}^{M}$ as representatives of the stored feature values. The features we use here have the fixed ordering $f_1, f_2, \ldots, f_M$ (cf. §2). If no stroke satisfies any given feature i, then $f_i := 0$. The feature vectors of all the N (learning sample) sketches are collected in the feature space F. If $f_i \neq 0$, then the selected feature number i is satisfied by exactly f_i strokes (of each of the N sketches). The value $f_i \neq 0$ means that either (i) f_i represents the number of geometric shapes (for the corresponding relative size features), (ii) f_i represents the number of pairs of geometric shapes (for positional relation features), or (iii) f_i represents the number of n geometric shapes (for similarity group $n \geq 2$). Thus, the QR feature space F is a matrix of $N \times (M+1)$ integers, in which the N rows represent the N sketches, while its first M columns represent the M QR feature vectors of the sketches, and the $(M+1)^{\text{st}}$ column is a class value (e.g., the sketched object's label; cf. the "class value" column in the right part of Figure 2).

1.2 Example: The $\mathcal{S}_{w_1}$ wheel

The wheel in Figure 1a has been sketched in the following order: outer circle/ellipse, inner ellipse, four lines started at down right and move clockwise. For this object, our technique extracts several features, such that the

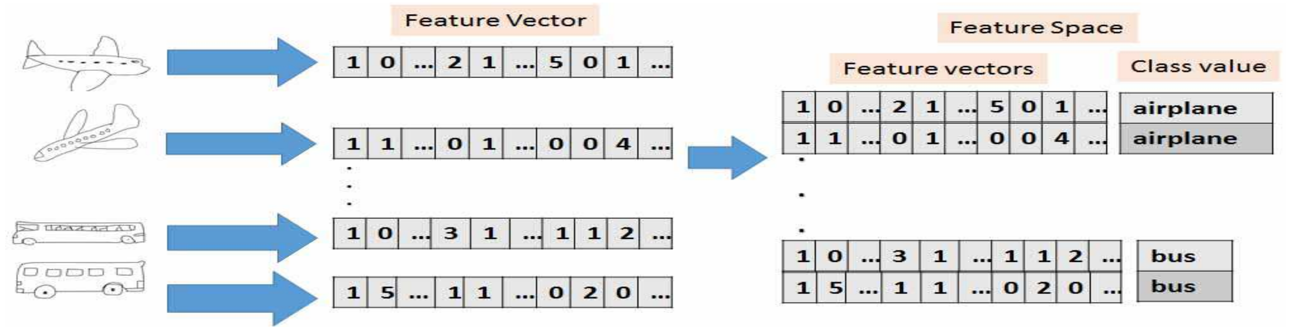


Fig. 2: Constructing the Features Space F from the N sketches.

relative size of the outer ellipse is medium, the smaller ellipse is tiny, and the four lines are tiny. The positional relations between the outer circle and the inner circle is that the inner is equal to the outer (w.r.t. centroid positions), but the first line is down right of the inner ellipse, the second line is down left the first one, etc. The four lines are grouped together and create one group (cf. §1.1). Let $\mathcal{S}_{w_1} = \langle s_i \rangle_{i=1}^{7}$ represent the wheel sketch in Figure 1a. Let s_1 and s_2 be the outer and inner "circles", respectively, and assume that their centroids have the same location. Let s_3 and s_4 be the right and left strokes, respectively, below the dashed line in the same figure (that is, s_4 is the line stroke immediately below the inner circle s_2, and s_3 is the stroke to its right in the figure[1]). Instead of the postfix notation, $\equiv (s_2, s_1)$, of the positional relation "equal", we use the infix notation $s_2 \equiv s_1$ to indicate that s_2 is located at the same location of s_1. Since s_3 is a successor to s_2, and spatially located to the latter's "down-right", we write $s_2 \searrow s_3$. Similarly, $s_4 \swarrow s_3$ indicates that the semi-vertical line segment, s_4, comes "down-left" of (and successor to) the leaned line segment, s_3. The five consecutive line strokes $\langle s_i \rangle_{i=3}^{7}$ of the wheel $\mathcal{S}_{w_1}$ of Figure 1a are successive and have the same relative size (which is "tiny" in this case). Therefore, these strokes are combined, forming a group labelled "*group of three or more lines*". The feature vector for $\mathcal{S}_{w_1}$ contains values that reflect the following features of the wheel in Figure 1a: five tiny lines, one tiny circle, and one small circle. This means that the value of feature "tiny-line" is 4, of "tiny-circle" is 1, and of "small-circle" is 1. The similarity group labelled "*group of two circles*" has the value 1.

2 Methodology

We present a Qualitative Representation used for Sketch Recognition Technique, or *QuRSeR-Tech* for short, which is proposed to build its classification model by learning a restricted set of common qualitative representations. QuRSeR-Tech is used for recognizing newly drawn sketches based on a previously given set of learning sketches. QuRSeR-Tech consists of two learning systems and, consequently, two kinds of datasets used for learning purposes.

The first classifier utilizes HOG-based extracted features for recognizing drawn strokes within **geometric shapes**. The geometric shapes are considered the low-level representation constituents of a sketch. Here, these geometric shapes are "line", "arc", "ellipse", "poly-line", "triangle", "rectangle", or "polygon". For each geometric shape, we collect a set of hand-drawn sketches describing it. These shapes are used as a training dataset in order to learn and build a classifier model that, in turn, is able to classify a given stroke as one of the geometric shapes. To that aim, we extract the features that discriminate each kind of image. The first dataset contains hand-drawn strokes (the learning samples), each of which is classified into one of the geometric shapes. The second classifier uses QR-based extracted features for recognizing a hand-drawn sketch to an object name. The second dataset contains hand-drawn sketches, each of which is classified into one of the **object names**: "AIRPLANE", "BICYCLE", "BUS", or "HOUSE".

QuRSeR-Tech's first learning system is called *StrokeToGeometric-classifier* (StoG), which builds its classifier model to recognize a hand-drawn stroke (as one of the aforementioned geometric shapes). After that, all recognized geometric shapes and their qualitative representations are stored together, to build a new dataset that will be set as input for the second learning system, which we call *SketchToObject-classifier* (StOb). The latter is a backbone of QuRSeR-Tech, and is is used for classifying a hand-drawn sketch.

[1] Note that the labeling of the strokes is intended to reflect the successiveness of the strokes; e.g., s_4 is sketched after s_3. This is vital, especially because the presented positional relations are not all commutative.

2.1 StoG classifier: The 1^{st} Learner

The Histogram of Gradient Orientation (HOG) is a method that is based on evaluating well-normalized local histograms of image gradient orientations in a dense grid. HOG's basic idea is that local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or edge directions, even without precise knowledge of the corresponding gradient or edge positions [Dalal et al., 2006].

In order to extract HOG features of an image, one starts with describing the structure of the image from which the HOG features are extracted. Each gray image is represented by a matrix $\texttt{IMG}_{w\times h}$ of integer numbers, w and h, varying between 0 and 255, with $\texttt{IMG}(i, j)$ representing the intensity of the image at pixels $1 \leq i \leq w$ and $1 \leq j \leq h$. The image $\texttt{IMG}$ is resized to $\texttt{IMG}'_{128\times128}$. The function, *extractHOGFeatures*[2], is applied to extract the HOG features of the resized image $\texttt{IMG}'$, which gives the ability to set the values of its parameters depending on the needs. The output of this function is the extracted HOG features from a gray image $\texttt{IMG}'_{128\times128}$, which returns a 1-by-250 feature vector. These features encode local shape information from regions within an image. Therefore, for each image, we extract 250 features stored as an instance in the dataset. As a result, if we have k images, then we extract a feature space as a matrix called $\texttt{HogF}_{k\times250}$, with one extra column attached that represents the class value (geometric shape) of each image. So the output is $\texttt{HogF}_{k\times251}$.

A support vector machine (SVM) [Scholkopf and Smola, 2001] is a supervised learning algorithm capable of building a classifier model by learning from training samples to analyze data used for classification. In our case, SVM employs the extracted feature space $\texttt{HogF}$ (of all images as training samples for the learning process), to produce a classifier model, called StrokeToGeometric-classifier (StoG), which is able to classify a drawn stroke as one of the geometric shapes mentioned earlier. In order to use StoG for testing an unknown classified image $\texttt{UIM}_{w\times M}$, we go through the following steps: (i) resize the image into $\texttt{UIM}'_{128\times128}$, (ii) extract its HOG features *UIMHOG* which will be a vector 1×250, (iii) apply StoG to UIMHOG, where the output will be the class value (geometric name) of this image. Figure 3 gives an example of a real case of a hand-drawn sketch (BUS), in which there are 15 strokes that have been drawn. By applying StoG to each one of the drawn strokes, we are able to classify them into: polygon, ellipse, ellipse, ellipse, line, line, line, ellipse, line, line, line, rectangle, rectangle, rectangle, and rectangle.

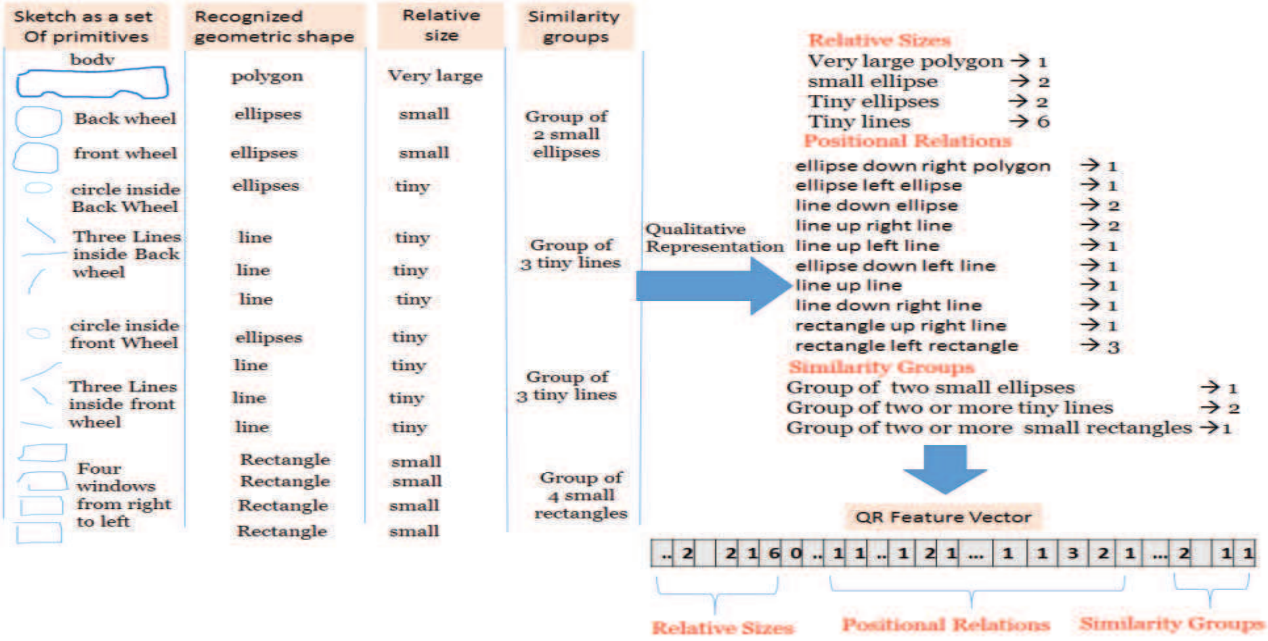


Fig. 3: QR feature vector of the sketched bus, $f_{\mathcal{S}_{b_1}}$, of Figure 1b.

[2] This is one of the predefined functions within the Matlab computer vision toolbox.

2.2 Qualitative Representation (QR) Identification

Our main goal in this paper is to study the effect of some qualitative representations on building an accurate classifier model that is able to recognize hand-drawn sketches. We present three qualitative representations, along with an argument why they may be considered as suitable candidates. Recall that each sketch $\mathcal{S}$ consists of a set of $m(s)$ strokes, classified into one of the mentioned geometric shapes.

Relative sizes of strokes:

The absolute size of a stroke $s := \langle p_i \rangle_{i=1}^{n(s)}$, called $size(s)$, is the accumulated distances between every two successive points p_i and p_{i+1} belonging to s (for $i < n(s)$). This can be formulated as $size(s) = \sum_{i=1}^{n(s)-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}$. The relative size is also calculated for each stroke w.r.t. the sketch it belongs to. In our case, we use only relative sizes based on the discretization of the absolute size into one of the following seven discrete values: tiny, small, medium, semi medium, large, very large, huge. We conducted experiments with human subjects, in which the participants draw wheels of busses in different sizes, but almost always approximate a relative size (say "medium") w.r.t. the whole bus sketch. Furthermore, also based on our experiments, rectangles appear (as windows) in the sketches of a house object in large or very large relative sizes, whereas rectangles (as windows) in the sketches of bus objects appear in medium relative sizes (w.r.t. the whole bus sketch). Figure 3 shows that the relative sizes of the drawn strokes: polygon, ellipse, ellipse, ellipse, line, line, line, ellipse, line, line, line, rectangle, rectangle, rectangle, and rectangle are very large, small, small, tiny, tiny, tiny, tiny, tiny, tiny, tiny, tiny, small, small, small, and small, respectively.

We extracted 49 relative size features; for each one of the seven geometric shapes, we use seven relative size features (tiny, small, ..., etc.). Therefore, the relative size features are: tiny lines, ..., huge lines; tiny arcs, ..., huge arcs; ...; tiny polygons, ..., huge polygons. In Figure 3, the relative feature "very large polygon" is 1, which means that there is only one very large drawn polygon, and the relative size feature "tiny lines" is 6, which means that there are 6 drawn tiny lines.

Positional relations:

A positional relation is calculated based on geometric properties of every two successive strokes, s_i and s_{i+1}, where the relation is calculated for s_{i+1} with respect to s_i (cf. §1.1). Thus, we make use of 441 features: $s_i \; R \; s_{i+1}$, where $R \in \{\equiv, \rightarrow, \leftarrow, \uparrow, \nwarrow, \nearrow, \downarrow, \swarrow, \searrow\}$ (i.e., $7 \times 9 \times 7 = 441$). An example is shown in Figure 3, where the positional relation called "rectangle ← rectangle" is set to 3, because there are three pairs of successive strokes having the same positional relation feature.

Similarity groups:

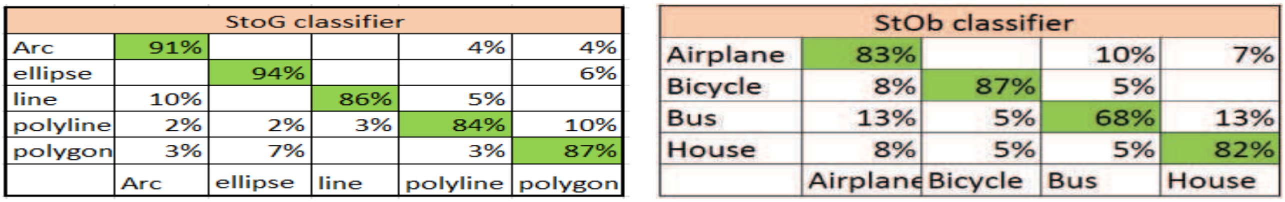
A similarity group is based on the human vision system, and aims to create groups of similar successive strokes. For instance, a participant drew many lines —of similar relative sizes— inside the wheel. These lines are grouped together to form one group, which may contain two geometric shapes or a group of more than two geometric shapes. Hence, we extracted 14 features; each one of the 7 geometric shapes s will have the corresponding 2 kinds of the similarity groups (group of tow geometric shapes or group of three or more geometric shapes). The similarity groups are groups of two lines, groups of three or more lines, ..., group of two polygons, groups of three or more polygons.

2.3 StOb classifier: The 2^{nd} Learner

The three kinds of extracted features for each sketch are stored together to form a matrix called QR feature space F (as mentioned before in §1). which are 504 (i.e., 49 relative sizes, plus 441 positional relations, and 14 similarity groups).

3 Experimental Results and Discussion

We aim at building a classifier model able to recognize a hand-drawn sketch into object name. To achieve this goal, two datasets have been collected through two experiments in order to provide our classification processes with many training datasets from which our learning system can learn and build its model. To evaluate our classifiers, we use a k-fold cross-validation technique with $k = 10$.



StoG classifier

	Arc	ellipse	line	polyline	polygon
Arc	91%			4%	4%
ellipse		94%			6%
line	10%		86%	5%	
polyline	2%	2%	3%	84%	10%
polygon	3%	7%		3%	87%

(a)

StOb classifier

	Airplane	Bicycle	Bus	House
Airplane	83%		10%	7%
Bicycle	8%	87%	5%	
Bus	13%	5%	68%	13%
House	8%	5%	5%	82%

(b)

Fig. 4: (a) Stroke to Geometric shape Classifier's Confusion Matrix. (b) Sketch to Object Classifier's Confusion Matrix.

1. *Hand-drawn Geometric Shapes Dataset:* In the first dataset, we collect many hand-drawn strokes that represent the constituents of any hand-drawn sketch and extract their HOG features to build a HOG feature space. These strokes are classified into one of the following geometric shapes: line, arc, ellipse/circle, polyline, triangle, rectangle, or polygon. Regarding this issue, a Matlab GUI interface has been implemented for sketching and collecting hand-drawn geometric shapes. Using this interface, the experiment was conducted on 6 subjects where each subject sketched all 5 geometric shapes with different scales, orientations, and positions. We collected 250 hand-drawn geometric shapes and converted them into HOG feature space. The core point is to apply one of the machine learning algorithms such as support vector machine SVM on the HOG feature space, and get a classifier named StoG classifier that will be used for recognizing any sketched stroke and classifying it to only the following geometric shapes line, arc, ellipse, poly-line, polygon. StoG classifier records accuracy= 88.4% for recognizing any hand-drawn geometric shape. Further mathematical analysis are made to identify the polygon to one of the following: triangle, rectangle, or polygon.
   Figure 4a shows the StrokeToGeometric-Classifier's confusion matrix, in which the arc and ellipses can be recognized with accuracy greater than 90%, while the other shapes can be recognized with accuracy greater than 83%. We note that 10% of the poly-lines are recognized as polygons, so we use further processes to decide whether the shape is open or closed.[3]
2. *Hand-drawn Sketches Dataset:* In this dataset, we collect many hand-drawn sketches for four specific objects: AIRPLANE, BICYCLE, BUS, and HOUSE. By implementing a matlab GUI interface, a second experiment has been conducted on 12 subjects where each subject drew 20 sketches —five for each one of the four objects. The qualitative representation QR of all those sketches are extracted and stored in QR feature space, F. By applying SVM to the feature space, F, the sketchToObject-classifier classifies a given hand-drawn object as one the mentioned 4 objects with 80% accuracy. Figure 4b shows the StrokeToGeometric-Classifier's confusion matrix, in which all objects are classified correctly with percentage greater than 80%. In order to show the effects of each QR feature (relative size, similarity group, and positional relation), we build and measure the accuracy of the following 7 models: relative size, similarity group, positional relation, relative size and similarity group, relative size and positional relation, similarity group and positional relation, and all QR. Figure 5 shows that for each one of the 7 cases, 4 classifier model are tested (linear SVM without using PCA for feature selection, linear SVM using PCA, Quadratic SVM without PCA, Quadratic SVM with PCA). The results show that the three combined QR feature plays an important role to achieve the best accuracy. One of the interesting notes is that, without using positional relation, we still have an accurate classifier. This implies that this feature needs more modifications.

All experiments have been implemented and tested on Windows 10, Matlab 2016a, computer vision toolbox, and the classification learner application. All sketches have been collected via WACOM pen, WACOM touch screen, and some sketches from the dataset in [Eitz et al., 2012] are used as guiding images to the participants.

4 Conclusion

Understanding what hand-drawn sketches represent, in a way similar to what humans do, is a challenging problem for computation. The representation of sketches using qualitative features plays a key role in this understanding. Qualitative models that can be used to represent sketches are mainly based on two aspects. The first aspect is the way of segmenting an object's contour into meaningful edges. The second aspect is the level of detail for describing an object. The way of handling these aspects can distinguish one model from another.

[3] E.g., the distance between the start- and end- points of the drawn stroke is compared to the length of the drawn stroke.

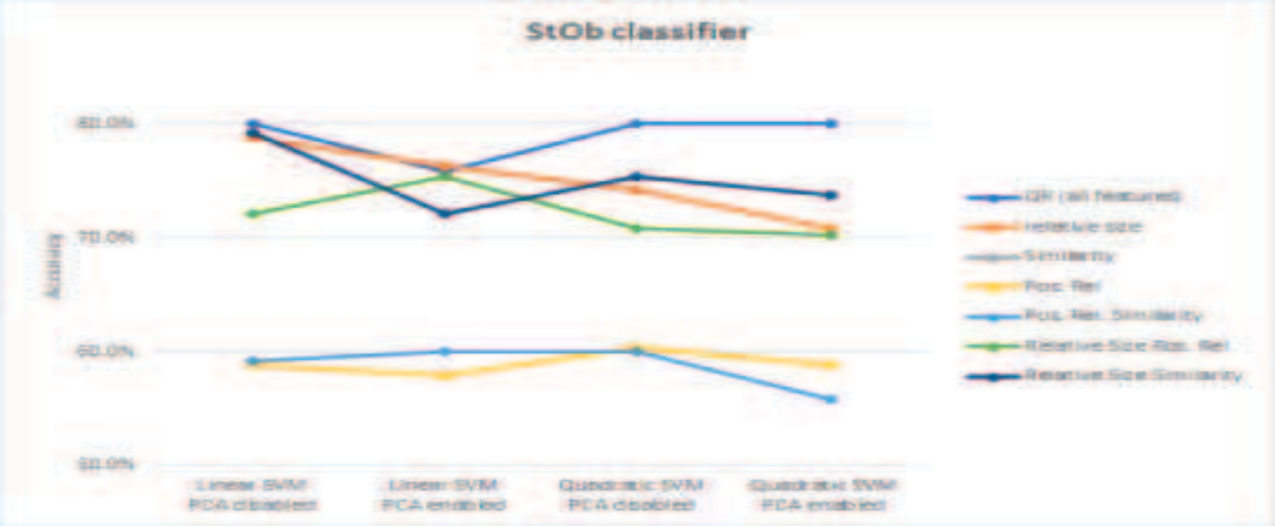


Fig. 5: SketchToObject classifiers accuracy

Here, a technique called *QuRSeR-Tech* has been proposed and implemented for recognizing hand-drawn sketch. Our technique consists of two main stages: StoG classifier (that uses the function extractHOGFeatures for extracting the HOG features of each stroke), and quadratic support vector machine SVM (that is based on the HOG features for recognizing the drawn stroke to geometric shape). To get an accurate classifier, we set the parameters cell size, block size, block overlap, number of bins, and the signed orientation to [4 4], [2 2], [50% 50%], 8, and false respectively. The StoG classifier recorded 88.4% accuracy for recognizing the strokes to geometric shapes. Second, the StOb classifier builds a new representation for any hand drawn sketch by recording three kinds of qualitative representation: relative size, positional relation, and group of similarity. Based on this new representation and quadratic support vector machine, the StOb classifier is able to recognize hand drawn sketches with accuracy 80%. Some comparative studies have been conducted to see the effect each one of the three kinds of QR, in which we proved that the StOb classifier's accuracy will be decreased if any of the mentioned representations is missing. In both of the two classifiers, we apply PCA to reduce the features to only the features that are highly correlated to the class value.

The ideas of this paper are in connection with others in the literature (such as [Wang et al., 2013], [McLure et al., 2011], [Paulson and Hammond, 2008], and [Lovett et al., 2006], to mention a few). [Wang et al., 2013] gives a method based on fuzzy hybrid-based features to classify strokes into geometry primitives. Their human computer interactive system is developed for determining the ambiguous results and then revising the missrecognitions, where their system is developed to classify eight primitive shapes. PaleoSketch is also a primitive shape recognizer that classifies single strokes into certain primitive geometric shapes (cf. [Paulson and Hammond, 2008]). [McLure et al., 2011] describe a higher-level of qualitative representation based on *edge-cycles*, which are sequences of edges connected end-to-end, whose last edge connects back to the first edge. The results of the edge-cycle representation outperforms that of the edge-level representation —produced by CogSketch in [Lovett et al., 2006]— in learning to classify hand-drawn sketches of everyday objects. Although the remarkable performance of edge-cycles representation, it has its own set of problems that can lead to misclassification of sketched concepts (cf. [McLure et al., 2011]).

References

[Cohn and Renz, 2008] Cohn, A. G. and Renz, J. (2008). Qualitative spatial representation and reasoning. *Foundations of Artificial Intelligence*, 3:551–596.

[Dalal et al., 2006] Dalal, N., Triggs, B., and Schmid, C. (2006). Human detection using oriented histograms of flow and appearance. In *European conference on computer vision*, pages 428–441. Springer.

[Eitz et al., 2012] Eitz, M., Hays, J., and Alexa, M. (2012). How do humans sketch objects? *ACM Trans. Graph.*, 31(4):1–10.

[Lovett et al., 2006] Lovett, A., Dehghani, M., and Forbus, K. D. (2006). Efficient Learning of Qualitative Descriptions for Sketch Recognition. In *Proceedings of the 20 th International Workshop on Qualitative Reasoning.*

[McLure et al., 2011] McLure, M. D., Friedman, S. E., Lovett, A., and Forbus, K. D. (2011). Edge-cycles: A qualitative sketch representation to support recognition. In *Proceedings of the 25th International Workshop on Qualitative Reasoning*.

[Paulson and Hammond, 2008] Paulson, B. and Hammond, T. (2008). Paleosketch: Accurate primitive sketch recognition and beautification. In *Proceedings of the 13th International Conference on Intelligent User Interfaces*, IUI '08, pages 1–10, New York, NY, USA. ACM.

[Scholkopf and Smola, 2001] Scholkopf, B. and Smola, A. J. (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

[Wang et al., 2013] Wang, S., Wang, G., Gao, M., and Yu, S. (2013). Using fuzzy hybrid features to classify strokes in interactive sketches. *Advances in Mechanical Engineering*, 5:259152.