MOBI-COG: A Mobile Application for Instant Screening of Dementia Using the Mini-Cog Test

Shahriar Nirjon, Ifat Afrin Emi, Md Abu Sayeed Mondol, Asif Salekin,
and John A. Stankovic
Department of Computer Science
University of Virginia
{smn8z, iae4qb, mm5gg, as3df, stankovic}@virginia.edu

ABSTRACT

In this paper, we present MOBI-COG which is an application that runs on a mobile device, such as a tablet or a smartphone, and provides an automated and instant dementia screening service. The MOBI-COG App is a complete automation of a widely used 3-minute dementia screening test called the Mini-Cog test, which is administered by primary caregivers for a quick screening of dementia in elderly. Besides asking the patient to remember and then recall a set of three words, the test involves a free-hand clock drawing test. The MOBI-COG App automates all these steps – including the automatic assessment of the correctness of a clock drawn on the touch screen of a mobile device. We train the MOBI-COG App with over 1000 touch-drawn clocks and show that the system is capable of detecting and recognizing digits in less than 100 ms, in-situ (i.e. without the help of any back-end server), with 99.53% accuracy, and is robust to changes in people, sizes of the drawn digits, and screen sizes of the mobile devices. We perform a usability study of MOBI-COG involving eight healthy human subjects and show that the system is capable of performing all three steps of the test effectively. We also provide a summary of the users' comments on the application.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]: Real-time and embedded systems.

General Terms

Design, Experimentation

Keywords

Dementia, Mini-Cog Test, Cognitive Screening

1. INTRODUCTION

More than 5 million Americans today are living with various forms of dementia. Among them, 60% – 80% are suffering from Alzheimer's – which is the deadliest form of dementia [3]. A systematic and regular screening of cognitive impairment enables early detection of dementia and allows patients and their families to make important decisions regarding transportation, living arrangements, and other aspects of care, such as activation of diagnosis, treatment, and support services when the patient is functioning at his highest possible level [22, 8]. Screening dementia in a clinical setting requires taking the patient to a primary care facility. This is often pushed back in negligence as it involves planning, and costs time, effort, and money. A convenient, automated, and easy to administer dementia screening test which can be taken at home on a day-to-day basis is thus an attractive way of primary screening of cognitive impairment. In this age of wireless health where in-home health monitoring is becoming a reality, such a system is highly desirable.

With this goal in mind, we introduce MOBI-COG[1] which is an application that runs on a mobile device, such as a tablet or a smartphone, and provides an automated and instant dementia screening service. The MOBI-COG application is a complete automation of a well-known dementia screening test called the Mini-Cog test [5, 20, 6]. The test is widely used by primary caregivers for a quick screening of dementia in patients. The intended use case of MOBI-COG is to let an elderly take the test using a large-screen tablet device where he is posed with a set of words to remember, followed by a clock drawing test, and finally a test to recall the words. At the end of the test, the system analyzes the recalled words and the clock drawn by the user on the touch screen, and provides a score as done in a paper-based Mini-Cog test. A family member should be administering the test in case the patient is not capable of taking it alone or is using the application for the first time. Note that, this application does not give any dementia assessment. It only automates the paper-based standard test. The relationship between the test-scores and the level of dementia must be determined by an expert doctor or a caregiver as done in the paper-based tests. Hence, the application should not be used as an alternative to seeing a caregiver; rather it is desirable that the history of test results, which is stored into the system, is shared with the caregiver for a better understanding of the patient's condition.

Several salient features when taken in combination make MOBI-COG unique. First, this is a mobile application that

[1]A video demo of the App can be found at [1].

automates a dementia screening test involving technical challenges such as an automatic evaluation of the correctness of a touch-drawn clock on a mobile device. Other existing applications either offer simple questionnaire-based tests or just provide information on dementia to create awareness. Second, the application is fast. It provides test results in less than 100 ms and the test scores are deterministic when compared to a human caregiver. Caregivers take time in scoring the tests and their scoring is subject to human errors. Third, the system is accurate. The recalled words are never miscounted and the automatic assessment of the clock drawing test is also highly accurate. Fourth, the system is lightweight. It performs all its computation in-situ, i.e. without the help of any remote server and hence, Internet connectivity is not a requirement. Fifth, the system is very easy to use. A family member of the patient with zero clinical knowledge or even a patient with mild cognitive disorder can administer the test without needing any assistance.

We have implemented the MOBI-COG App on Android OS. The core technical challenge in our implementation has been the detection and recognition of digits and hands on a clock drawn on the touch screen. We solve this by implementing a k-NN classifier which mainly uses chain-codes [13] as features. We conduct three sets of experiments to demonstrate the performance of MOBI-COG. First, we measure the CPU and memory footprints of the application. Second, we evaluate the accuracy and classification time of the k-NN classifier. Third, we perform a usability study of the complete MOBI-COG App by having eight healthy volunteers perform all three steps of the test and then rate and comment on various usability aspects of the system.

The contributions of this paper are the following:

- MOBI-COG, a fast, accurate, light-weight and easy-to-use mobile application that automates the Mini-Cog dementia screening test and provides a convenient way of monitoring early symptoms of memory impairment.
- An implementation of a k-NN classifier that is capable of automatically assessing the correctness of a clock drawn on the touch screen of a mobile device. Trained on over 1000 training examples, the recognizer is shown to be capable of detecting and recognizing touch-drawn digits and hands on a clock in less than 100 ms, with an accuracy of 99.53%.
- A dataset having 1026 touch-drawn clocks on two different sizes of touch screen devices (two tablets and a smartphone). The dataset and a program to read it are downloadable from [1].

2. THE MOBI-COG APPLICATION

The MOBI-COG App is a complete automation the Mini-Cog dementia screening test. The App can be used as a fast and effective tool for screening dementia in patients with cognitive disorder – with or without the help of a caregiver. The App runs on a mobile device that has a touch screen such as a tablet or a smartphone. The user of the App performs all three steps of the Mini-Cog test interactively, and the App provides an automated assessment at the end of the test.

Figure 1 shows four screenshots of the MOBI-COG App after a user has taken the test. The first three screenshots correspond to the three tasks in a Mini-Cog test and the last one shows the summary of the test result. These four steps are described next.

2.1 Task 1 - Remembering Words

The first task for the patient in a Mini-Cog test is to remember a set of three words. The user is shown three randomly chosen words from a local database on the mobile device and is prompted to read aloud and remember them. Since the application is supposed to be used by an elderly, the font size of the words to remember is made bigger than usual. The collection of words is created by taking words from example tests that we found online and in the literature [9, 10, 17]. Figure 1(a) shows an example where the user is shown and asked to remember the words {`Leader, Season, Table`}.

2.2 Task 2 - The Clock Drawing Test

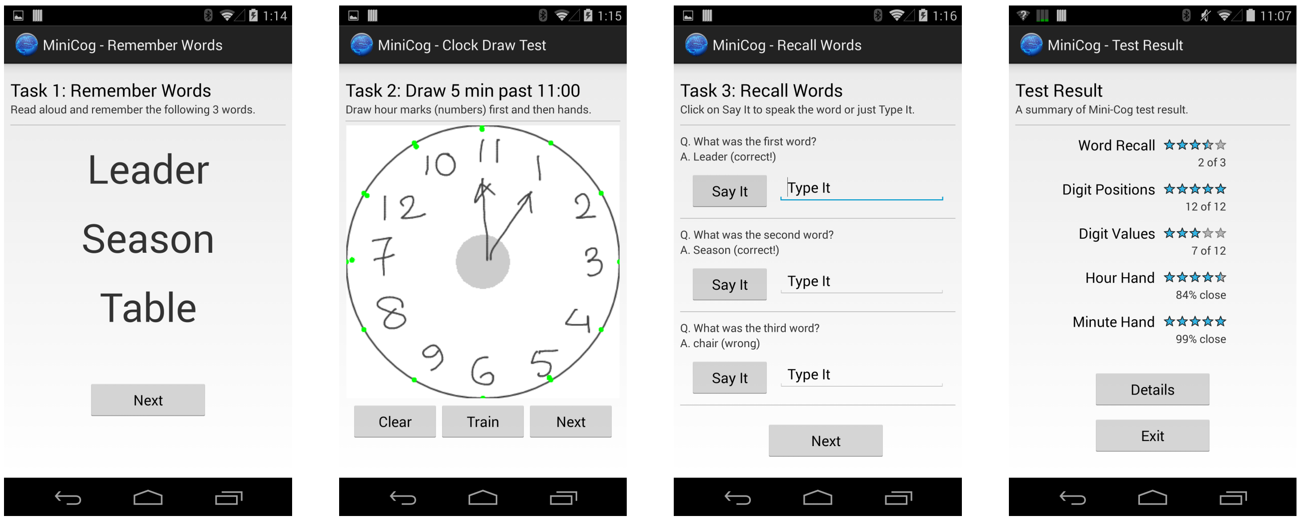
The second task for the patient is a clock drawing test. In the pen and paper version of this test, the patient is given a piece of paper with a circle drawn on it and is asked to draw the digits and hands of the clock showing a given time of the day. The patient is supposed to write each of the twelve numbers of a clock near the appropriate hour mark position and to draw hands showing the given time. The duration of this step is three minutes and hence, this test is often called the 3-minute clock drawing test (CDT).

In MOBI-COG, we automate this by showing a large circle (whose diameter equals to the width of the screen) on the screen and asking the patient to draw the numbers and hands to show a given time. The time is chosen randomly and is of the form 'X minutes past Y', where X is kept a multiple of 5 for simplicity. As the patient draws the digits and the hands using his finger on the touch screen, an algorithm running on the background identifies the digits and their intended position and computes several correctness measures, such as the correctness of positioning the digits, the correctness of the sequence of the digits, and the closeness of the drawn hands. The algorithmic details of the process is described in the next section.

In Figure 1(b), the user draws 12 numbers on the clock and of which, only 7 are in the right position. Of the two clock hands, the minute hand is pretty close to the 5 min mark, whereas the hour hand is about 30 degrees off the right position.

2.3 Task 3 - Recalling Words

The third task for the patient is to recall the three words that he was shown in step 1. To the caregivers, this is the



(a) Task 1 (Remember) (b) Task 2 (Clock Draw) (c) Task 3 (Recall) (d) Results

Figure 1: Screenshots from the MOBI-COG application showing completed tasks and the test result.

most important step as most dementia patients would forget at least two out of three words after taking the clock drawing test. In a traditional Mini-Cog test, a patient utters or writes down the recalled words on a piece of paper and a caregiver makes an assessment. In MOBI-COG, the user is prompted by the system to speak the word aloud. The system uses the built-in speech-to-text engine of the mobile device to infer the word and matches it to the corresponding word from the first task. As it is sometimes hard to infer the right word from the speech, especially when the person is a non-native English speaker, MOBI-COG also comes with an option of text input. Figure 1(c) shows that our user uses the speech option to input the words and recalls two out of three words correctly.

2.4 Test Result

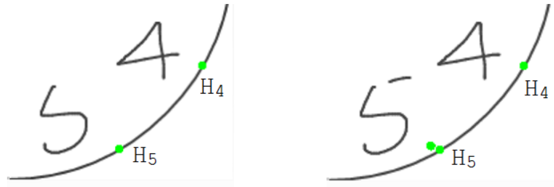
The MOBI-COG App provides a summary of the test results as well as the details. The summary, as seen in Figure 1(d), shows (1) word recall – which is the number of words that the user recalled correctly, (2) digit positions – which denotes how many of the twelve positions had any digit drawn, (3) digit values – denoting how many of the digits were drawn correctly at the right position, and (4) hour and (5) minute hands – denoting the closeness (angular distance) of the two clock hands to their correct positions. This information is shown graphically using five 5-starred rating bars along with numerical values. The detailed result, on the other hand, shows the details of each of these, i.e. which words were recalled correctly and which were not, which of the twelve numbers were missing or misplaced, and the exact amount of angular deviations of the drawn clock hands.

For our example scenario, Figure 1(d) shows that the word recall was 66% accurate as two out of the three words were correctly recalled, each of the twelve hour marks (i.e. 100%) had a digit drawn near it, seven out those twelve digits (i.e. 58.3%) were placed at the right position, the hour hand was about 30 degrees off (i.e. 16%), and the minute hand was almost 100% close to the exact location.

3. THE CLOCK DRAWING TEST

The clock drawing test is the center piece of the MOBI-COG App where a clock drawn on the touch screen of the mobile device is automatically analyzed for correctness. Technically, this is similar to handwriting recognition (HWR) or optical character recognition (OCR) problems where alphanumeric characters on an image are recognized. But some properties of the clock drawing test on a touch screen make it easier than the generic OCR problem and open up room for a better performance.

There are several open-source software and services [7, 21, 2] that are capable of recognizing handwritings with a reasonable accuracy. However, compared to those general-purpose OCR engines, our problem is much simpler as we require recognizing only ten digits as opposed to recognizing a long sequence of alpha-numeric characters. In addition to that, running a heavy-weight and computation-intensive OCR engine on a mobile device requires more memory and is extremely slow, as the App has to convert the drawing to an image before processing, and image manipulation is extremely slow on a mobile device. A touch screen, on the other hand, provides us with a sequence of 2D points (typically $30-50$ points per digit) as the user draws a digit on the screen. Processing such a small amount of data directly on the main memory is fast. The knowledge of the order of the points also saves a large chunk of computation time as finding this order is often the first and an expensive step in OCR. Hence, instead of using an OCR system, we implement our own algorithm which is highly accurate and also lightweight.



(a) After 1^{st} stroke (b) After 2^{nd} Stroke

Figure 2: The digit 5 is drawn with two strokes.

An alternative to recognizing touch-drawn clocks could be to let the user drag and drop already drawn digits and hands. We did not do this as it would alter the way the Mini-Cog test is taken.

The process of recognizing digits on a clock involves primarily three steps: (1) associating a digit to a specific hour mark, (2) recognizing the digit, and (3) identifying the hands.

3.1 Associating Digits to Hour Marks

The touch screen interface provides a sequence of 2D points starting from the moment when the user first touches the screen and ending when he lifts his finger up. We call such a sequence a 'stroke'. A digit may consist of multiple strokes depending on how the user draws it. The first step in MOBI-COG is to assign each stroke to one of the twelve hour marks, which is done as follows.

Step 1: Assign each hour mark a variable 2D coordinate $H_i(x, y)$, where $1 \leq i \leq 12$. $H_i(x, y)$ is initialized to the coordinates where the i-th hour mark is placed on any clock. For each stroke S_k, repeat the following two steps:

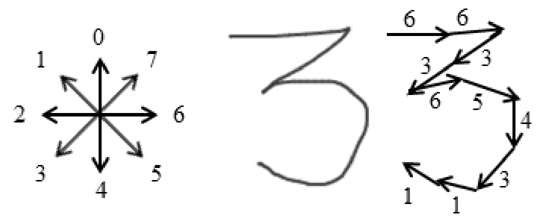
Step 2A: Compute the center of mass of S_k, $COG(S_k)$ and associate it with the hour mark for which the Euclidean distance $||S_k - H_i||$ is minimum. Let's say it is H_{min}.

Step 2B: Update H_{min} to $0.85 \times H_{min} + 0.15 \times COG(S_k)$.

Figure 2(a) shows a partially drawn clock where the user has just finished drawing the first stroke of the digit 5. The system computes its closeness and associates it with H_5, and updates the coordinates of H_5. The updated coordinates of H_5 is shown on Figure 2(b) as a second dot near the 5-hour mark. The closeness of the next stroke, which is the rest of the drawn digit 5, is now computed using this new H_5 mark. The reason we update the coordinates of the hour marks is that, on a device having a smaller screen-size (5 inches or less), often the correct hour mark for a stroke is farther than a nearby hour mark. Moving the hour mark closer to the partially drawn digit's center of mass helps reducing this distance. For a device with a larger screen-size (7 inches or more), this is however is not an issue.

3.2 Recognizing Digits

Once we have identified the set of strokes for each of the twelve hour marks, we start recognizing digits by processing each set individually. This is a three step process: (1) preprocessing, (2) feature extraction, and (3) classification.



(a) Eight directions (b) Computing chain-codes

Figure 3: An illustration of chain-code for the digit 3.

3.2.1 Preprocessing

Each stroke of a digit goes through a normalization step. The goal of which is to make sure that the size of a drawn digit does not affect the recognition process and the aspect ratio is preserved. In order to do so, the following two steps are performed in order:

Step 1: All the points on a stroke are translated so that its center of mass becomes the origin. This is done by subtracting the mean of all X-coordinates (Y-coordinates) from each point's X-coordinate (Y-coordinate).

Step 2: The range of X-coordinates and the range of Y-coordinates are computed. Each point is scaled so that the larger of the two ranges is mapped to $[-100, +100]$. This ensures that the digit is normalized and the aspect ratio is also preserved.

3.2.2 Feature Extraction

We compute a 19-element feature vector for each stroke. If a digit has multiple strokes, the feature vectors are element-wise added to obtain a single feature vector.

The first eight elements of the feature vector are the eight-directional chain-codes [13]. The chain-code is a simple yet highly effective feature which is widely used in hand-written character recognition problems. Given a sequence of points in 2D, the N-directional chain-code is an N-bin histogram where each bin corresponds to a range of directions in 360° and contains the number of vectors, joining two consecutive points on a stroke, whose directions fall into the range.

An illustration of chain-code computation is shown in Figure 3. Figure 3(a) shows eight equally spaced angles in 360° on a plane. Figure 3(b) on its right shows a touch-drawn digit 3 along with the vectors formed by two consecutive points on it. Each of the vectors is assigned a chain-code based on the closeness of its slope to one of the eight directions. We intentionally have made the length of the vectors larger in this figure for the clarity of viewing. The computed chain-code of the digit is `(0,2,0,3,1,1,3,0)`.

The next eight elements of the feature vector is another set of eight-directional chain-codes. These are similar to the previous ones except for this time, instead of taking consecutive two points, we skip a point and take the next to the next point when computing the direction. We do this to make the feature robust to short-term variations on a drawn digit.

This additional set of chain-codes is more effective when the mobile device has a higher screen resolution and skipping points while computing the directional vectors provides a better approximation of the curvature of the digit.

The next element of the feature vector is an indicator of whether or not it is a multi-digit number. We determine this by computing the bounding boxes of each set of strokes and testing whether or not one box is completely left (or right) to the other one. We take this additional feature into our feature vector in order to simplify our implementation, so that we do not have to invoke the digit recognition routine twice for the two-digit numbers. The first 16 features, i.e. only the chain-codes, are not good enough to distinguish between 11 and 1 as their codes are similar. Hence, having this extra feature greatly helps eliminating such confusions and improves the accuracy.

The last two elements of the feature vector are the fractions of total points on a digit having positive X and positive Y coordinates, respectively. These two features encode the symmetry of a digit with respect to the X and Y axes, and they help distinguish digits like 6 from 9, who have similar chain-codes but are different in symmetry with respect to the axes.

3.2.3 Training and Classification

We use a k-nearest neighbor (k-NN) classifier to recognize the digits. We have chosen k-NN for its simplicity of implementation. The algorithm does not require any model training. It simply matches an unknown example with each of the training examples, finds the closest k matches, and then performs a majority voting. This makes it easier to implement on a mobile device and also easier to train further. In MOBI-COG, we use Euclidean distance between two vectors as the distance metric and choose $k = \sqrt{n}$, where n is the total number of training examples.

The MOBI-COG App comes with a pre-loaded set of training examples. Each training example is a 19-element feature vector which is extracted from a correctly drawn clock and is stored inside the file system of the device. Although MOBI-COG comes with a pre-computed training set, a new example can also be added to the set. The App allows the user to save a correctly drawn clock as a training example for use in later tests.

3.3 Identifying the Clock Hands

The two hands of a clock drawn by the user are also stored as two sets of strokes. We distinguish them from the digits by the closeness (linear distance) of their one end to the center of the clock. The direction of a hand is computed by joining the tip of a hand to the center of the clock and then taking its slope. In MOBI-COG, we ignore the length of a hand and distinguish between the hour and the minute hands by their closeness (angular distance) to the respective hand on a clock in which the given time has been drawn correctly.

4. EVALUATION

	CPU	Memory
Nexus 5	3% (19%)	32.2 MB (48.1 MB)
Nexus 7	4% (13%)	21.5 MB (39.9 MB)
Galaxy Tab 3	10% (27%)	14.9 MB (43.3 MB)

Table 1: CPU and memory footprints.

We describe three types of evaluations. First, we measure the CPU and memory footprints of the App. Second, we evaluate the accuracy of clock recognition. Third, we perform a usability study of the complete system.

4.1 Experimental Setup

We have used three mobile devices in our experiments. Two of them are tablets having a 7 inches touch screen. One of which is a Nexus 7 (quad-core 1.51 GHz) and the other one is a Samsung Galaxy Tab 3 (dual-core 1.2 GHz). The third device is a Nexus 5 (quad-core 2.3 GHz) smartphone having a 5 inches screen.

We train the k-NN classifier using the training data collected from a total of 7 participants (3 females and 4 males). All participants are healthy and their ages are in the range of 25 – 35. They have diversities in writing-style and speaking-style. The participants are asked to draw each of the twelve numbers on a clock at an appropriate position and in their natural way. The dataset contains 1026 training examples for each digit and can be downloaded from [1].

4.2 CPU and Memory Footprint

We measure the CPU and memory footprints of MOBI-COG when the App is running. We use Android Debug Bridge shell's (adb shell) `top` command to measure the CPU and memory usages. Table 1 shows the average (and the maximum in brackets) CPU and memory usages of the MOBI-COG App for all three models of the mobile devices. The average CPU utilizations in Nexus devices having quad-core CPUs are negligible ($\leq$ 4%). The Galaxy Tab consumes more CPU cycles on average as it has a dual-core CPU. The peak CPU utilization may go up to 13% – 27%, but this happens for a short duration (100 ms) when the device is performing the k-NN search. Hence, the battery consumption by this application is negligible. The maximum memory usage is about 50 MB, which is due to loading all the training examples into the memory. Such an usage of memory is comparable to applications' such as Maps (62.7 MB) and Music (54.9 MB). The total size of the binary is only 556 KB.

4.3 Evaluation of the Clock Recognizer

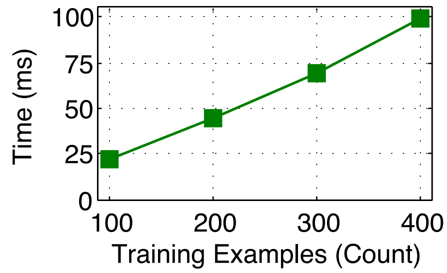
4.3.1 Accuracy

The goal of this experiment is to quantify the accuracy of the k-NN digit recognizer. We train the classifier with 1026 correctly drawn clocks where each digit were drawn at its

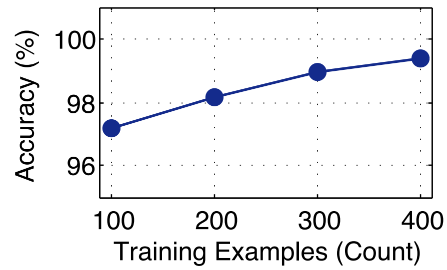
Prediction outcome

Actual Value	p	n	Total
p′	True Pos 4,619	False Neg 133	4,752
n′	False Pos 133	True Neg 52,139	52,272
Total	4,752	52,272	

Table 2: Confusion matrix of the digit recognizer.



(a) Classification time vs. Training



(b) Accuracy vs. Training

Figure 4: Classification time increases linearly with training examples, and the accuracy crosses the 99% **mark for** 300 **or more examples.**

right position. During testing, we create a separate set of test cases which has 396 touch-drawn clocks (or $396 \times 12 = 4752$ numbers). This set contains four types of test cases: complete clocks with no errors, partial clocks with no errors, complete clocks with errors, and partial clocks with errors. Table 2 shows the overall confusion matrix of the digit recognizer. As each positive example acts as a negative one for all other classes, the total number of negative examples is 11 times of the positives. The accuracy of the recognizer is 99.53% with a precision and recall of 97.2%. The recognizer makes mistakes for a small number of cases with a false positive and false negative rate of 0.5%. By carefully analyzing those examples we see that the most difficult cases were the ones that involved {2, 3, and 5}.

4.3.2 Classification Time

As the *k*-NN classifier compares an unknown example with each training example, the classification time dependents highly on the size of the training set. A larger training set increases the classification time, but at the same time it increases the classification accuracy. So there is a trade off between these two. Figure 4 quantifies this trade-off by showing the accuracy and classification time for various sizes of training sets. We use the Galaxy Tab tablet in this experiment as this is the slowest of the three devices. The Nexus devices being faster, classification time on these devices is less. The accuracy of the digit recognizer does not depend on the model of the device.

Figure 4(a) shows that the classification time increases almost linearly with the size of the training set. The classification time is less than 100 ms as long as the number of training examples is 400 or less. Figure 4(b) shows that the classification accuracy also grows with more training examples, but once the system has seen 300 or more examples, the accuracy crosses the 99% mark and thereafter it does not increase much. Hence, 300 – 400 training examples are sufficient to classify the digits in 100 ms, with an accuracy of over 99%. If we choose to use a faster device, such as the quad-core Nexus devices, the classification time is even less. In these devices, MOBI-COG is capable of handling 1026 training examples in less than 100 ms and achieves 99.53% accuracy.

4.4 Usability Study

We perform a usability study of MOBI-COG in which, we ask eight volunteers (four males and four females) to take the complete Mini-Cog test using the App. Each user takes the test twice, first using a tablet and then using a smartphone. As all the participants are healthy, we do not expect them to draw incorrect clocks or forget the words. This is why we asked four of the volunteers to deliberately make mistakes while drawing the clock or recalling the words. At the end of the test, they participate in an online survey where we ask them questions regarding the usability of the system. The usability questionnaire is not any standard set of questions rather from our own curiosity. Table 3 shows the summarized result of the survey. The average score for each question is shown on it's right. The scores are in a scale of $1-10$, where 10 is the best. The outcomes of this experiment should not be generalized as it is biased by the age and ICT knowledge level of our participants.

Although this was a small scale survey, yet the results are interesting. The first two questions were asked with the expectation that everyone would be in favor of a larger screen. While most of them rated the larger screen higher, one of our participants thinks that the smaller device is much handy. However, she wrote in the comments that it was her personal opinion and for the elderly she would suggest a larger screen. We got mixed results for the input methods too. Half of the participants hated the speech input as the device wouldn't understand their accent and they had to speak the same word

	Question	Score
Q1.	How comfortable were you with drawing the clock on a smartphone compared to pen and paper?	7.5
Q2.	How comfortable were you with drawing the clock on a tablet compared to pen and paper?	8.0
Q3.	To what extent you would be comfortable if the App only had the speech input?	6.5
Q4.	To what extent you would be comfortable if the App only had the text input?	7.6
Q5.	To what extent do you believe that your App-based test scores will be the same as in paper-based tests?	9.0
Q6.	To what extent do you believe that the App is usable for dementia screening in a clinical setup?	9.1
Q7.	Rate the UI of the App. Suggest any modifications you would like to see in comments.	9.1
Q8.	Rate your overall satisfaction on the App. Provide the details in comments.	8.5

Table 3: The questionnaire and the result of the usability study.

multiple times to get it right. This is one limitation of the Google's speech recognition engine. We investigated this further and found that the device that was running an older version of the Android OS (Android 4.1) was more troublesome. We would thus recommend anyone using the MOBI-COG App to upgrade their OS to the latest version (Android 4.4) for a better speech recognition.

The fifth and the sixth questions were about the acceptability of the App as an alternative to the paper-based test. Our participants seemed extremely happy with the App and would recommend the App for a replacement of the paper-based test without any reservation. The overall satisfaction of the App was rated to 8.5 out of 10. Those who rated lower brought up some interesting suggestions, such as keeping a provision for a stylus-based input so that it is easier to draw, and one claimed that it is possible to cheat with the App as the App lets the user go back to the first screen and check what the words were in the first task.

5. DISCUSSION

All the participants in our usability study belonged to the age-group of 20 – 40 and none of them had a history of any kind of cognitive impairment. An elderly with cognitive conditions are likely to make mistakes while drawing a clock. To be able to automatically detect those cases, during our evaluation, we design the test cases (Section 4.3.1) in a manner that the set contains both correct and incorrect clocks as well as complete and partial ones. Our evaluation shows that the MOBI-COG App is capable of detecting all such cases with an accuracy of 99.53%.

A healthy person may draw a clock differently than an elderly. We may expect differences in speed, pressure, and size and shape of the finger. This is however not an issue in MOBI-COG as the k-NN classifier used in digit recognition is generic as opposed to person specific. The features used by the algorithm are solely related to the characteristics of the drawn digits and does not depend on any person specific features such as the speed, pressure, or size of the finger. As long as a drawn number shows resemblance to any of the twelve numbers, the system is capable of recognizing it accurately.

Devices having a 5 – 7 inches screen may not be large enough for some elderly to feel comfortable with. For those, we would recommend a 12.2 inches tablet which is the largest Android tablet available in the market today. The MOBI-COG App supports multiple screens and runs on Android devices of all screen sizes without any modification. As the drawn digits are normalized before feature extraction, the accuracy of the digit recognizer does not depend on the size of the screen. Some elderly may not be comfortable with touch screen devices. In such cases, a suitable alternative is to use a stylus-based tablet device which is closer to a pen and paper based test setup.

Using the application by the user every now and then could potentially have a learning effect. To alleviate this to some extent, we have made the words and the time that appear on a test random. However, we suggest that the user (or the family member who administers the test) should follow the guidelines of his caregiver to know the prescribed interval between successive tests.

Participants in our usability study believe that the App is usable for dementia screening in a clinical setup. This however is not sufficient as we are not really sure how acceptable the system will be to a caregiver and/or to a patient. We do not evaluate the validity of a touchscreen-based test as an alternative to a paper-based test in this paper. With regard to this, we would argue for its benefits, such as convenience, uniform scores, immediate results, ease of administering a test, and the ability to keep history; we would recommend that the system be used for primary screening at home and not as an alternative to periodic visits to the caregiver; and we would hope that, as more and more such wireless healthcare systems emerge and become wide-spread, people will eventually accept systems such as this.

6. RELATED WORK

There have been several dementia and Alzheimer's screening tests for use in general medical practice. The basic principle in all these tests is somewhat similar to Mini-Cog's – consisting of a controlled learning step, followed by a short delay, and then recall. Buschke [9] proposed the Memory Impairment Screen (MIS) test where the subject is given a set of four words from four different categories to remember and later asked to recall them with and without the category cues. Das [10] proposed the DrD Quick and Easy (Q&E) dementia screening test where subjects are given three pairs of words to remember followed by simple tasks (e.g. remembering the date, and verbal fluency test), and then recalling the words without any cue. Mendiondo [17] proposed a formal scoring system for screening mild Alzheimer's disease (AD) which is a weighted summation of the scores from four tests: a three word recall test, a date remembering test, a spelling a word test, and a naming animals in 30 seconds test.

There are a few mobile Apps related to dementia and cognitive health. There is one category of Apps in Google Play that only provide information on dementia to create awareness. Examples include – Dementia Symptoms, Dementia Support, Dementia Care, and Signs of Dementia. There is another category of Apps that helps people with memory impairment to remember things (Remember First), to remember time (Alzheimer's Dementia Day Clock), with hands free calling (Alzheimer Phone), and to navigate back to home (AlzNav). But none of these are for screening dementia. A third category of Apps, e.g. Dementia Screener and MMSE for Alzheimer Disease, performs questionnaire-based dementia screening where the App asks questions based on Hodkinson [14] and AD8 [12, 18]. Compared to these, MOBI-COG is more sophisticated in terms of technicality and completeness.

There are several online and offline algorithms for recognizing hand-written characters. [19] provides a generic survey of several such algorithms and [23] provides a survey on the online algorithms only. The algorithm that we use in MOBI-COG is online where digits are recognized as they are drawn. We use a k-NN classifier in MOBI-COG as it is the simplest to implement on a mobile device and does not require any training phase other than just storing the training examples. However, there are works that use more sophisticated classifiers, such as Hidden Markov Models (HMM) [15], Support Vector Machines (SVM) [4], and Artificial Neural Networks [16], for handwritten character recognition and pen drawn digit recognition [11]. Although these classifiers are excellent in recognizing characters, our k-NN classifier is sufficient in terms of both accuracy and classification time given the need of our application.

7. CONCLUSION

This paper presents the design, implementation and evaluation of a mobile application called the MOBI-COG App. The application is a complete automation of a well-known dementia screening test namely the Mini-Cog test. Compared to a pen and paper based test, the benefits of MOBI-COG include convenience, uniform scoring, instant results, ease of administering and taking a test, and the ability to keep history of test scores. This application should be used for day-to-day primary dementia screening in a home environment, and should not be taken as an alternative to seeing the caregiver. Our evaluation shows that the application performs all three steps of the Mini-Cog test effectively and the system is capable of assessing the correctness of a clock drawn on the touch screen of a mobile device in less than 100 ms and with 99.53% accuracy.

Acknowledgement

This paper was supported, in part, by NSF Grants CNS-1319302 and CNS-1239483, and a gift from PARC, Palo Alto.

8. REFERENCES

[1] MOBI-COG Project. http://tinyurl.com/pdatogd.
[2] OCR Web Service. http://www.ocrwebservice.com.
[3] A. Association et al. 2014 alzheimer's disease facts and figures. *Alzheimer's & Dementia: The Journal of the Alzheimer's Association*, 10(2):e47–e92, 2014.
[4] C. Bahlmann, B. Haasdonk, and H. Burkhardt. Online handwriting recognition with support vector machines-a kernel approach. In *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*, pages 49–54. IEEE, 2002.
[5] S. Borson, J. Scanlan, M. Brush, P. Vitaliano, and A. Dokmak. The mini-cog: a cognitivevital signs measure for dementia screening in multi-lingual elderly. *International journal of geriatric psychiatry*, 15(11):1021–1027, 2000.
[6] S. Borson, J. M. Scanlan, P. Chen, and M. Ganguli. The mini-cog as a screen for dementia: validation in a population-based sample. *Journal of the American Geriatrics Society*, 51(10):1451–1454, 2003.
[7] G. Bradski. *Dr. Dobb's Journal of Software Tools*.
[8] C. Brayne, C. Fox, and M. Boustani. Dementia screening in primary care: Is it time? *Jama*, 298(20):2409–2411, 2007.
[9] H. Buschke, G. Kuslansky, M. Katz, W. Stewart, M. Sliwinski, H. Eckholdt, and R. Lipton. Screening for dementia with the memory impairment screen. *Neurology*, 52(2):231–231, 1999.
[10] P. Dash. The q&e (quick and easy) 2.5-minute dementia screening test. In *ANNALS OF NEUROLOGY*, volume 52, pages S68–S68, 2002.
[11] K. Felch. Integrating digitizing pen technology and machine learning with the clock drawing test. *MIT*, 2010.
[12] J. Galvin, C. Roe, K. Powlishta, M. Coats, S. Muich, E. Grant, J. Miller, M. Storandt, and J. Morris. The ad8 a brief informant interview to detect dementia. *Neurology*, 65(4):559–564, 2005.
[13] R. C. Gonzalez and R. E. Woods. Digital image processing, 2002.
[14] H. Hodkinson. Evaluation of a mental test score for assessment of mental impairment in the elderly. *Age and ageing*, 1(4):233–238, 1972.
[15] J. Hu, M. K. Brown, and W. Turin. Hmm based online handwriting recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(10):1039–1045, 1996.
[16] B. B. Le Cun, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*. Citeseer, 1990.
[17] M. S. Mendiondo, J. W. Ashford, R. J. Kryscio, and F. A. Schmitt. Designing a brief alzheimer screen (bas). *Journal of Alzheimer's Disease*, 5(5):391–398, 2003.
[18] J. C. Mundt, D. M. Freed, and J. H. Greist. Lay person–based screening for early detection of alzheimer's disease development and validation of an instrument. *The Journals of Gerontology Series B: Psychological Sciences and Social Sciences*, 55(3):163–170, 2000.
[19] R. Plamondon and S. N. Srihari. Online and off-line handwriting recognition: a comprehensive survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):63–84, 2000.
[20] J. Scanlan and S. Borson. The mini-cog: receiver operating characteristics with expert and naive raters. *International journal of geriatric psychiatry*, 16(2):216–222, 2001.
[21] R. Smith. An overview of the tesseract ocr engine. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition - Volume 02*, ICDAR '07, pages 629–633, Washington, DC, USA, 2007. IEEE Computer Society.
[22] P. R. Solomon and C. A. Murphy. Should we screen for alzheimer's disease? *Geriatrics*, 60(11), 2005.
[23] C. C. Tappert, C. Y. Suen, and T. Wakahara. The state of the art in online handwriting recognition. *IEEE Transactions on pattern analysis and machine intelligence*, 12(8):787–808, 1990.