Stroke Detector - An Application that applies the F.A.S.T. Test to Identify Stroke

Mihnea-Ioan Rezmeriţă, Irina-Elena Cercel, Adrian Iftene
Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iaşi, Romania
{mihnea.rezmerita, irina.cercel, adiftene}@info.uaic.ro

DOI: 10.37789/rochi.2020.1.1.7

ABSTRACT
In Romania, stroke is the second leading cause of death and disability (Neagu, 2019). The speed of reaction in the case of a stroke can make the difference between life and death, between a healthy patient and a patient who remains disabled for life. In this context, we wanted to create a mobile application in Romanian that will help users to do a F.A.S.T. test very quickly. The test identifies possible problems at the face, arms and speech levels. This article shows how we made this application using a client-server architecture and what users think about it.

Author Keywords
F.A.S.T., Client-server architecture, Stroke, Face detection, Speech recognition.

ACM Classification Keywords
H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces. H.3.2. Information Storage and Retrieval: Information Storage.

General Terms
Human Factors; Design.

INTRODUCTION
We now live in a world where technological progress makes our lives easier and better. We live in an era where almost anything is possible, an era in which there are no restrictions of any kind, and software and hardware development amazes us from day to day. Mankind has progressed extremely fast in the last 20 years, and fields such as medicine have evolved a lot.

Medicine is more than vital to us because it directly contributes to our quality of life. For this reason, medicine is one of the sciences most often approached by researchers in all fields, in order to improve existing tools. Over the past 20 years, researchers have been able to successfully solve some of the most important problems and mysteries in the medical field, eradicating various diseases and reducing the risks of certain diseases that were once considered lethal. A crucial role in the development of medical systems was played by the field of informatics. Computer science, like other sciences, has benefited from an astonishing growth, which is why the fields that relied on it (including medicine) have developed even more. So computer science has helped on the one hand to finalize tools to help medical subfields such as Medical Imaging, Epileptology, Oncology or on the other hand has helped to develop tools capable of influencing the ability of doctors in various other subfields.

This paper addresses a topical issue in the medical field, namely the detection of stroke. Even today, stroke is an extremely difficult problem for doctors because time is a decisive factor for the long-term evolution of the patient. In the case of stroke, the recognition of symptoms at an early stage, as early as possible determines whether or not the patient will be left with certain long-term sequelae (Weber, 1995). The current solutions address cerebral stroke detection and monitoring using cloud services (García et al., 2019).

WHAT IS STROKE?
Stroke is a very serious condition that endangers the proper functioning of the body and in many situations can lead to a disability or even lead to death[1]. It is found in all age groups, although mostly in the elderly. It is important to note that a crucial role in the occurrence of stroke is played by the medical history of the victim and also the medical history of first-degree relatives (Clarke and Forster, 2015).

General description
Stroke occurs when blood supply to a portion of the brain is interrupted (Marin, 2019). Discontinuation of blood supply for a long time causes the death of nerve cells in the affected section of the brain, leading to disabilities. In

[1] https://www.nia.nih.gov/health/stroke

many countries stroke[2] is considered to be the number one cause of disability. There are several types of stroke[3] (see Figure 1) (Sorenson et al., 2019):

- **Ischemic stroke** - accounts for approximately 87% of all stroke cases and occurs when an artery that irrigates nerve tissue is blocked.
- **Hemorrhagic stroke** - occurs when certain blood vessels rupture and cause internal bleeding. This causes a lot of pressure on the brain and causes a massive loss of blood in the surrounding areas.
- **Transient stroke**[4] - is also known as mini-attack and is caused by a temporary blockage of blood vessels leading to the brain. It should not be ignored, as it can be a potential symptom of a future ischemic stroke.

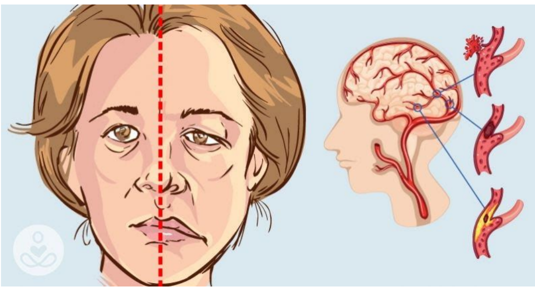

Figure 1: Stroke types[5]

Stroke can occur in a variety of forms and can vary from patient to patient. However, each patient will in turn experience one or more symptoms as well[6]:

- *Confusion* - generalized and cognitive problems;
- *Speech problems* - a visible difficulty in pronouncing words correctly;
- *Weakness* - some incapacity partial (or complete) to move a particular member;
- *Facial asymmetry* - it often happens that the mouth is dropped to one side;
- *Sudden and severe headache*.

Risk factors

Stroke is one of the leading causes of disability. However, it can be prevented by avoiding risk factors[7]:

- *High blood pressure*;
- *Smoking* - which is also a triggering factor in the event of other diseases;
- *Diabetes* - many people with diabetes have high blood pressure, are overweight or have high cholesterol;
- *Sedentary lifestyle* - which causes other problems;
- *Obesity* - which can still be combated with the help of a healthy lifestyle;
- *Heart disease* - of course, a medical history specific to heart disease is an important factor in the occurrence of stroke.

Forecasts and the importance of timely action

It is very important to act knowingly from the first sign of a stroke (Shugalo, 2019). We know that a stroke leads to the death of nerve tissues, so any wasted second is to the detriment of the patient.

Figure 2: FAST Test[8]

Some of those who suffer a stroke are left with sequelae, consequences, such as various paralysis or cognitive and speech problems (Brown, 2002). In order to be able to act in the shortest possible time, the test F.A.S.T.[9] was developed (see Figure 2), an abbreviation that aims to incorporate all the main symptoms:

- F - Face asymmetry;
- A - Arm weakness;
- S - Speech problems;
- T - Time to call the ambulance.

This paper presents the implementation of an application in Romanian that is able to perform an analysis similar to that performed by the F.A.S.T. to be able to act in time.

[2] https://www.stroke.org/en/about-stroke
[3] https://www.laboratorpraxis.ro/bine-de-stiut/tipuri-de-accident-vascular-cerebral
[4] https://www.stroke.org/en/about-stroke/types-of-stroke/tia-transient-ischemic-attack
[5] https://www.dornamedical.ro/images/articole_media/art_neuro_iunie_2019/POZA%20AVC%206.jpg
[6] https://www.medlife.ro/glosar-medical/afectiuni-medicale/accident-vascular-cerebral-avc-cauze-simptome-tratament
[7] https://www.stroke.org/en/about-stroke/stroke-risk-factors
[8] https://image.shutterstock.com/image-vector/stroke-warning-signs-symptoms-icon-260nw-1008179179.jpg
[9] https://www.stroke.org/en/about-stroke/stroke-symptoms

EXISTING SOLUTIONS

Fast Test

This application[10] was developed by CHHS[11], an organization that provides services to those affected by stroke and to raise awareness in Scotland about various general aspects of stroke. As seen in Figure 3 (in left), the colors are very appropriate, combining a bright pink with a cooler color.

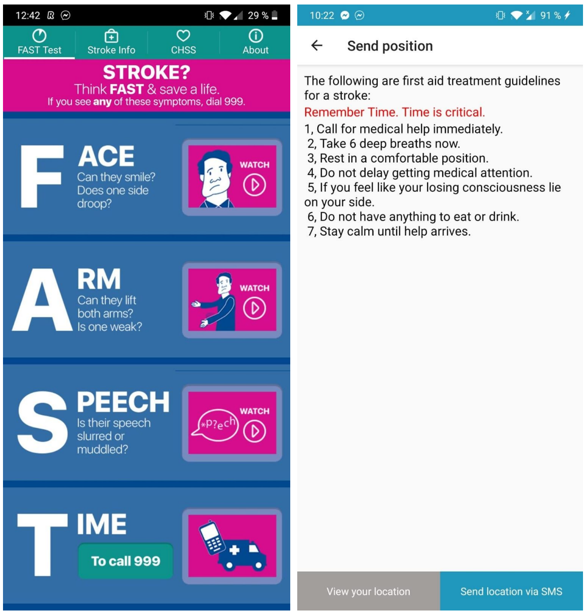

Figure 3: FAST Test App (in left) and S@S App (in right)

The application aims to present the details related to each component of the F.A.S.T. to those who install it. In addition to descriptions and suggestive images, the application also has short videos for presentation.

Although the application looks good and behaves just as well, it has one major drawback: it does not test the appearance of the stroke, but only presents it. We consider the presence of a component to perform an F.A.S.T. would have brought a very useful functionality to it.

S@S

Another application is SPOT@STROKE[12]. It was developed by user Kieran O'Callaghan to help people who are suspected of having a stroke. This project has an interesting approach based on a checklist that presents the main symptoms of stroke. The app also has the option to take a few photos relevant to the F.A.S.T. test, but does not process or use them to identify the stroke. A very good idea of the application is the presence of a close contact list. This list is very useful when we need to contact an acquaintance of someone who has such a stroke.

Another interesting component is the Send Position option that allows users to send the coordinates to which they are to a specific contact in the contact list (see Figure 3 in right). As with the previous application, there are some useful features, but it also lacks the F.A.S.T. attesting to the presence or absence of a stroke.

PROPOSED SOLUTION

The application we made is built on a server capable of detecting the most important features of the F.A.S.T test: facial asymmetry, arms weakness and speech problems.

System Architecture

The application uses a client-server architecture (similar to architecture from (Siriţeanu and Iftene, 2013)). The client sends requests to the server, which it receives, processes them, and finally sends the response back (see Figure 4).

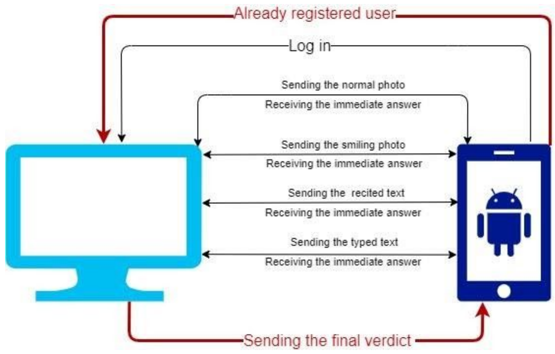

Figure 4: System Architecture

How the application works

In order to use the application, a user must create an account in advance. When he creates an account, he will go through several steps, such as taking a normal picture of him, taking an image in which he smiles, reading a text and typing a text. This information will form the user's profile and will be used as a reference when he wants to take an F.A.S.T. The F.A.S.T. involves following steps similar to those performed when creating the account:

- Take a normal picture of the user at the mobile device level and send it to the server. Its processing is done on the server, and it is

[10] https://play.google.com/store/apps/details?id=uk.org.stroke.fasttest
[11] https://www.chss.org.uk/
[12] https://play.google.com/store/apps/details?id=com.whereiskieran.android.spotamaster

compared with the profile image in order to detect the facial asymmetry specific to the stroke;

- Take a picture of the user smiling at the mobile device and send it to the server. On the server it is processed and compared with the profile image in which the user smiles in order to detect muscle weakness in the face;
- Retrieve user-readable text as an audio file at the mobile device level and send it to the server. The server compares the text extracted from the audio file with the text that the user had to read in order to detect speech problems;
- Retrieve a text typed by the user after a text received by him at the mobile device level and send it to the server. The server compares the initial text with the text typed by the user in order to identify the weakness of the arms.

At the end of the test the user will receive a warning in case he is suspected of a stroke and the necessary advice to help him manage the situation. If clear signs of a stroke are detected in one of the above steps, the test stops and skips to the last warning step and provides advice on how to handle the situation. At the client level there is also an information component, where the general symptoms of the stroke are presented, through a slide show.

Each component will be analyzed in more detail in the following chapters.

Server Component

The server was created using Python and Flask[13]. The server consists of several endpoints, all secure, so that they can only be accessed by authenticated entities. The server uses a database for user authentication, an authentication system, a local photo storage system, and various classes needed to parse the information and determine the final decision.

The server consists of several packages, files, modules and classes, each with a well-established role in stroke detection. Its purpose is to deal with requests submitted by users:

- login - Connecting and obtaining the necessary tokens, supports only POST type requests;
- register - User registration, supports only POST type requests;
- check_symmetry_normal_img - determination of facial features with the help; picture sent by the user, only supports POST requests;
- check_smiley_corners - determining the degree of muscle weakness in the face with the help of the picture sent by the user, supports only POST requests;
- get_text - returns a text to be typed or recited, supports only GET requests;
- parse_voice - determining the existence of speech problems using the text sent by the user, supports only POST requests;
- send_texting_test - determining muscle problems using text sent by the user, supports only POST requests;
- send_final_result - returns a final result, only supports GET requests.

Applications specific to the FAST test will return partial scores, which will be used to calculate the final score and make the final decision.

Android Client

The client is an application that runs on an Android device. The Android client offers an attractive and easy to use interface, including the F.A.S.T. and another special component for presenting symptoms. The user is presented at the beginning, the home page, which exposes the 2 options (Figure 5 in left):

- Conectare (Login) - in which the client is authenticated;
- Înregistrare (Registration) - through which he can create a new account.

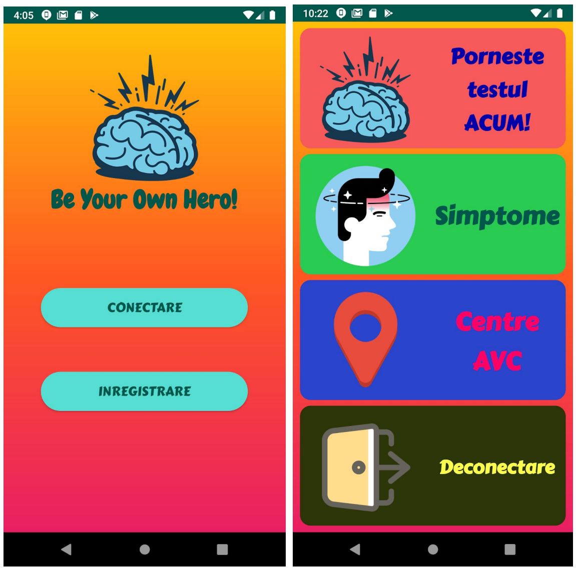

Figure 5: Home screen (in left) and Main menu (in right)

After connecting, the user is presented with the main menu, which allows several options, as shown in Figure 5 in right:

- F.A.S.T. test;

[13] https://flask.palletsprojects.com/en/1.1.x/

- Symptoms;
- Map of stroke centers (currently under construction);
- Disconnect.

FAST Test

The first step of the test is to send a normal picture of the test subject. He can choose from two options: either take a picture on the spot, or choose one from the gallery. The user is required to choose a picture in order to proceed to the next stage of the test. Once he sends the normal picture, in step two he is asked to send a picture in which he smiles (see Figure 6 in left).

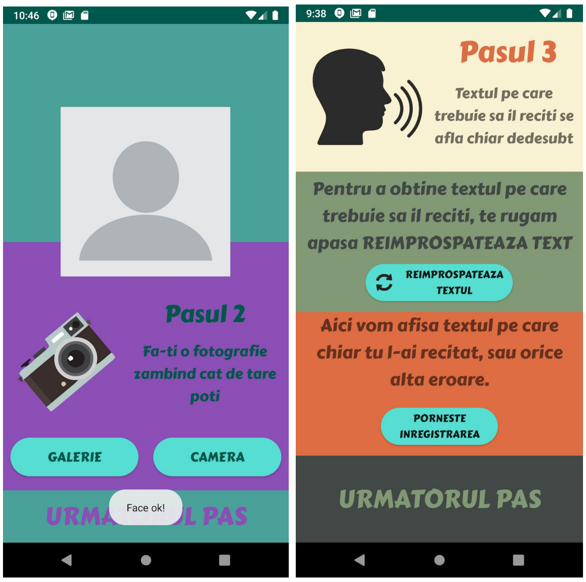

Figure 6: Step 2 (in left) and Step 3 (in right)

After these first two steps, the speech testing component follows (see Figure 6 in right). Here, the user's transition to the next stage is conditioned by 2 elements: obtaining the text to be recited and reciting it.

The user will initially get a text by pressing the *Refresh text* button. The required request will be called and the information obtained will be displayed. The second part of this step is the actual recitation of the obtained text, after pressing the *Start Recording* button, so that the device is aware that it needs to listen. Subsequently, the device will parse the recording in a very short time (approx. 2-3 seconds), and will display the resulting text. It is important to note that some audio recording parsing errors may occur, but they will be displayed so that the user understands the reason for the error.

The last step of the test is to detect the weakness of the arms, through a typing test. This test is similar to the previous one, as it consists of 2 components: obtaining the text, and typing it.

After completing this step, the user will receive the final verdict so that he knows if he is in danger or not. Depending on this we will receive some tips and suggestions, and the option to call a friend or even call 112.

Permissions

During the development of the client application it was important to consider the permissions, as the application needs to access various features specific to the Android platform. The application needs the user's consent for features such as:

- accessing the location;
- internet access;
- sound recording;
- access to the room.

Without these permissions, the F.A.S.T. cannot be achieved. Therefore, whenever needed, the user will be asked if it allows the application to use certain specifications. In case of a refusal, the functionality of the application is reduced, becoming almost useless. So, only if the user agrees to offer these permissions, the application can follow the course described above.

F.A.S.T. TEST

This chapter is responsible for explaining how we get facial features or how to achieve speech-to-text transformation..

Detection of facial features

One of the main components of the application is the detection of facial features. Detection of facial features is a problem often addressed by researchers, for which there are various viable solutions today (Asteriadis et al., 2011), (Barra et al., 2018), (Omer et al., 2019).

In our case, we used already predefined models and trained inside the dlib library[14] (Rosebrock, 2017). These models have a very good accuracy and are very effective in solving this problem[15] [16].

The chosen model was trained on the iBUG 300-W dataset [17], and determines 68 important points in front of a man, points that represent the following components (Barra et al., 2018):

---

[14] http://dlib.net/
[15] https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/
[16] https://www.geeksforgeeks.org/opencv-facial-landmarks-and-face-detection-using-dlib-and-opencv/
[17] https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/

- lip;
- eyebrows;
- eyes;
- jaw;
- nose.

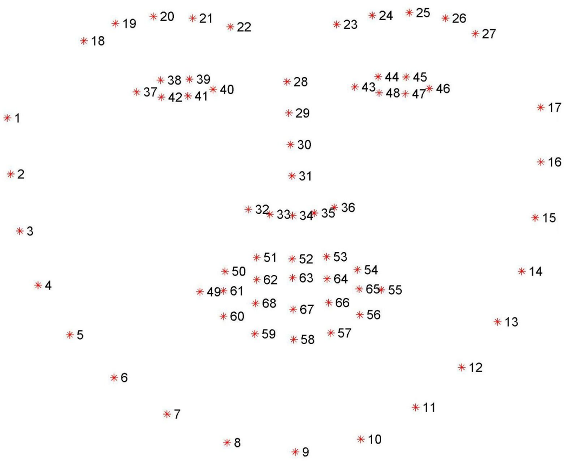

Figure 7: The 68 coordinates (Barra et al., 2018)

Also, the model used is a special model, already trained, which detects only the faces in the image, without a background. For this, the initial image is processed by the face detector, so as to restrict the area of interest in the picture. Having marked this area of interest in the picture, only it will be sent to the model that detects the specific elements of the faces. The algorithm after dialing will return a vector of 68 elements specific to the identified face (see Figure 8).

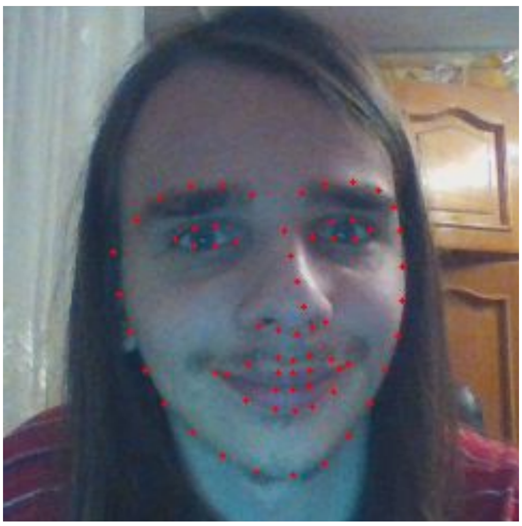

Figure 8: The 68 coordinates on an example

All these coordinates are part of a certain feature that we are looking for, for example, the mouth - the coordinates in the range [0-12], etc. Thus, the application determines the coordinates of these features.

Detection of face asymmetry

Knowing the above coordinates, the application is able to detect facial asymmetry. For the picture taken initially when setting up the user account and the current picture taken when performing the test, the positions of the eyes, the position of the mouth, the position of the corners of the mouth will be compared.

For the identified differences, an average of the coordinates on the left side of the mouth and an average of the coordinates on the right side of the mouth are calculated. Then the difference between the two parts is calculated. Similarly we do the same calculations for the eyes. These differences will give us a local score in the mouth, eyes, etc. Then with these local scores a global score will be calculated, which will help us in establishing the final verdict.

The most important local score is that of mouth asymmetry, which is associated with a common signal in the identification of stroke. Asymmetry of eyes, eyebrows, etc. represents secondary signals with less importance in the calculation of the final score.

Thus, if the mouth has a high degree of asymmetry, the application will decide that we have identified the stroke. If the asymmetry is average, the asymmetry of the eyes and other components will also be taken into account to calculate the final score. Also, a strong asymmetry in the eyes will also be sufficient for the detection of stroke, because the asymmetry of the eyes is also an important signal of the existence of stroke.

After several experiments and tests with the data we collected, we established the thresholds for the partial scores and for the final score, which told us that we have or have not identified a stroke attack in a patient.

Detection of muscle weakness in the face

Detecting muscle weakness in the face is not part of the F.A.S.T. test standard, but it is an approach that we have experienced in our tests and that could give good results in the future. To detect muscle weakness in the face, we check if the user is able to smile or not. For this, we compared the distances from the levels of the corners of the mouth. If the distance is above a certain threshold determined by us experimentally, then it means that the user does not have a problem. However, we must take into account the fact that some people smile differently, that is, they do not spread the corners of the mouth, but rather separate the lips so that the teeth can be seen. That's why we took this fact into account, calculating the distance between the upper and lower lip. If this distance is also too small, the user may have a problem.

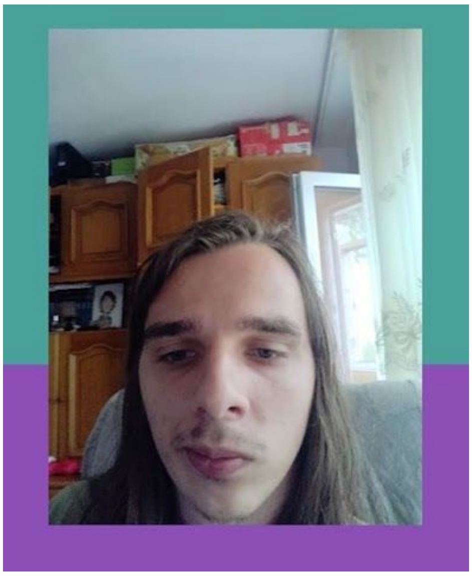

Figure 9: Smile attempt

To calculate these distances we also used the 68 coordinates from (Barra et al., 2018) and the images initially saved when the user created his account and the current image in which he smiles. Figure 9 is an attempt at a half-mouthed smile, which after the evaluation will notify the user that he is in danger.

Detection of speech problems

Another very important component of the test is the way in which speech problems are determined, similar to (Ţucă and Iftene, 2017). The user receives, at the client level, a text, which will have to be read by him. While the user is reading the received text, the client on the mobile phone will process the audio signal and turn it into text, using the component SpeechRecognizer[18] from Google. The resulting text will be sent to the server, along with the text received for recitation. The server will calculate the number of words that were mispronounced in relation to the text that was originally read.

Then, the number of mispronounced words will be used initially to calculate a partial score, and then to calculate a final score. Similar to the previous components, it can be decided based on the partial score whether we detected a stroke or not. The score threshold for determining whether or not we have a stroke was also identified after several tests and experiments.

Detection of muscle weakness in the arms

Muscle weakness in the hands is detected in a manner similar to that used in the component presented in the previous point. The user requests a text, which he will have to type, after which it will be sent together with the initial text to the server. If the user has problems with the hands, then he should not be able to type it.

At the server level, we used the Levenshtein distance (Levenshtein, 1966) to determine the differences between the two texts. These differences help us calculate a partial score for this component and then a final score.

After several experiments we determined that if the user makes at least half of the number of letters he had to type, then he is suspected of having a stroke.

The final decision

The final decision is the decision we make either based on a partial score that is above a certain threshold, or based on the final score, when it also has a value above a global threshold. We obtained all these thresholds experimentally after several tests and experiments that we did with several users..

USABILITY TESTING

Age categories

The application was tested by 22 people, in a direct way interacting with it but also indirectly by watching a video showing all the capabilities of the application. The interviewees are from different age groups as shown in the chart below.

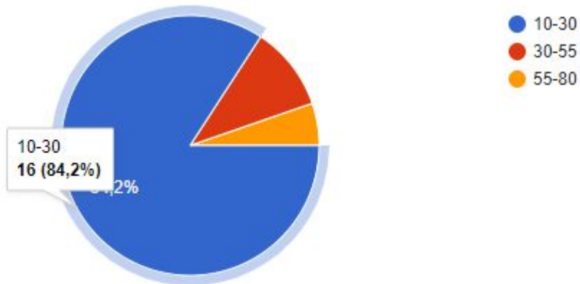

Figure 10: Age categories

The high percentage of people in the age category 10-30 is noticeable, but the other 2 categories should not be ignored either: 30-55 and 55-80 each with a percentage of 10.5% and 5.3% respectively. Everyone thought that the

[18] https://developer.android.com/reference/kotlin/android/speech/SpeechRecognizer

application was useful and that it could have a real impact on people's lives.

How intuitive is the application?

Many of the interviewees found the application intuitive and easy to use, but there were also people who reported that the application could be improved in terms of UI component and mode of operation, as reported in the chart below:

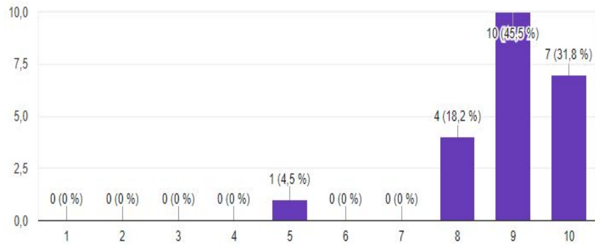

Figure 11: How intuitive is the application?

Regarding the waiting times, the opinions were mostly positive. Most felt that they did not have to wait long and others, fewer, considered that it took too long, especially when the Internet connection was not very good..

How do you like the design of the application?

From an aesthetic point of view, the design of the application was appreciated by users, they were pleasantly surprised. However, we believe that following the opinions gathered, it can be improved.

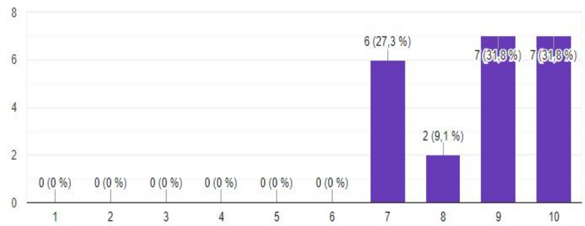

Figure 12: How do you like the design of the application?

All users considered that the application is of vital importance, and that they would use such an application if they were suspected of having a stroke. Among the most important improvements that users consider necessary are: the introduction of a machine learning component and the introduction of a voice narrator to guide users. Other improvements that could be made were the elimination of the recording component, the optimization of the symptom component and the introduction of voice commands.

Important future improvements

Also, many of the users interviewed preferred to express their opinion in a free way on things that could be improved. Among the things they listed are the following:

- Messages like "Call 112" or "We're sending the information to the server!" they should be placed somewhere at the top of the page so that the user can see the information more easily;
- Another suggestion would be to introduce more types of tests, the application contains the FAST test, but to make it more extensive, it could also contain other types of more complicated tests that test certain aspects in more detail, over a longer period of time, in order to be able to prevent a possible attack even before it occurs, to see the evolution in time of the user;
- If the elderly want to use the application for example, they may encounter problems such as: they do not understand the written instructions, the text is not large enough, they have difficulties in taking the picture. The idea with the vocal narrator is very good, but we consider that we have to repeat the commands when we do not have user interaction for a long time. We think a "Quick Test" option would be good (a quick test without the need for registration). Some font size settings or a preference for written / audio instructions would be helpful.

General opinion of the users interviewed

In general, users were satisfied with the initiative brought by the application, its purpose, how it works, how it behaves and the solution it proposes to solve this problem. The grades obtained by the application are in the range 7-10.

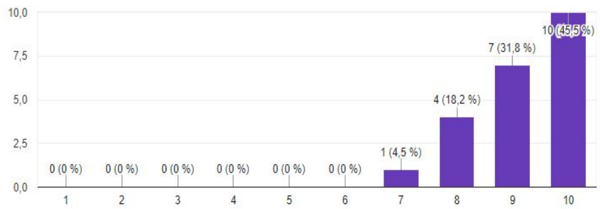

Figure 13: General opinion of the users interviewed

CONCLUSION

This paper describes an application capable of detecting the symptoms of a stroke. The app includes an F.A.S.T. digital, on the phone, so that it can be used by anyone. The F.A.S.T. it consists of 4 components: detection of facial asymmetry, detection of muscle weakness in the face, detection of speech problems and detection of muscle weakness in the hands.

The purpose of this application is to facilitate the diagnosis of stroke as soon as possible, in order to avoid possible problems involved in this disease. Currently, the duration of a stroke test using our application is about 2 minutes, but we intend to reduce this time even further in the future.

According to the reports and statistics made and presented in the chapter dedicated to testimonials, the integration of a narrator must be taken into account in the future, to guide users through audio instructions, through the application.

ACKNOWLEDGMENTS

This work was supported by project REVERT (taRgeted thErapy for adVanced colorEctal canceR paTients), Grant Agreement number: 848098, H2020-SC1-BHC-2018-2020/ H2020-SC1-2019-Two-Stage-RTD.

REFERENCES

1. Asteriadis, S., Nikolaidis, N., Nikolaidis, N., Pitas, I. A Review of Facial Feature Detection Algorithms. In book *Advances in Face Image Analysis: Techniques and Technologies*. 42-61 (2011), http://doi:10.4018/978 -1-61520-991-0.ch003
2. Barra, P., Bisogni, C., Nappi, M., Ricciardi, S. Fast QuadTree-Based Pose Estimation for Security Applications Using Face Biometrics. *Network and System Security. NSS 2018. Lecture Notes in Computer Science*, Springer, Cham. 11058 (2018) https://doi.org/10.1007/978-3-030-02744-5_12
3. Brown, M. M. Brain Attack: a new approach. *Clinical medicine (London, England)* 2(1): 60-5, (2002)
4. Clarke, D. J., Forster, A. Improving post-stroke recovery: the role of the multidisciplinary health care team. *Journal of Multidiscip Healthc*. 8: 433-442, (2015)
5. García, L., Tomás, J., Parra, L., Lloret, J. An m-health application for cerebral stroke detection and monitoring using cloud services. *International Journal of Information Management*, 45: 319-327, (2019)
6. Levenshtein, V. I. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8): 707-710, (1966)
7. Marin, C. Accidentul vascular hemoragic. *Doc.ro* (2019) https://doc.ro/accident-vascular-cerebral/accidentul-vascular-hemoragic
8. Neagu, A. 4 Romanians die every hour due to a stroke. Doctors are asking for funding and new centers to reduce the number of deaths. *Hotnews.ro* (2019)
9. Omer, Y., Sapir, R., Hatuka, Y., Yovel, G. What Is a Face? Critical Features for Face Detection. *SAGE Journals*, 48(5):437-446 (2019) https://doi.org/10.1177/0301006619838734
10. Rosebrock, A. Facial landmarks with dlib, OpenCV, and Python. *Pyimagesearch* (2017) https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/
11. Shugalo, I. How Artificial Intelligence Can Predict and Detect Stroke. *ITonline, Feature Artificial Intelligence*. (2019) https://www.itnonline.com/article/how-artificial-intelligence-can-predict-and-detect-stroke
12. Siriţeanu, A., Iftene, A. MeetYou - Social Networking on Android. *In Proceedings, of 11th International Conference RoEduNet IEEE: Networking in Education and Research*., Editor Rusu, O., 17-19 January 2013, Sinaia, Romania, 228-233, (2013)
13. Sorenson, T., Giordan, E., Lanzino, G. Neurosurgery for Ischemic and Hemorrhagic Stroke. *Fundamentals of Neurosurgery* (2019) https://doi.org/10.1007/978-3-030-17649-5_9
14. Ţucă, L. P., Iftene, A. Speech recognition in education. Voice Geometry Painter Application. *In the 9th Conference on Speech Technology and Human-Computer Dialogue*, *IEEE*, July 6-9, 2017, Bucharest, Romania, 1-8, (2017)
15. Weber, C. E. Stroke: Brain Attack, Time to React. *AACN Clinical Issues Advanced Practice in Acute and Critical Care* 6(4): 562-575, (1995)