
Pict2Text 2.0. Translating messages with pictograms into text



Final project
Course 2020–2021

Autors

Gasán Mohamad Nazer
and
Veronika Borislavova Yankova

Directors

Virginia Francisco Gilmartín
and
Susana Bautista Blasco

Bachelor's Degree in Software Engineering
Faculty of Informatics
Complutense University of Madrid

Pict2Text 2.0. Translating messages with pictograms into text

Bachelor's Degree Final Project in Software Engineering

Autors

**Gasán Mohamad Nazer
and
Veronika Borislavova Yankova**

Directors

**Virginia Francisco Gilmartín
and
Susana Bautista Blasco**

Convocation: *June 2021*

**Bachelor's Degree in Software Engineering
Faculty of Informatics
Complutense University of Madrid**

November 4, 2020

Contents

1. Introduction	1
1.1. Motivation	1
1.2. Goals	2
1.3. Software development methodology	2
1.4. Document structure	3

Chapter 1

Introduction

This chapter will display the motivation behind this project, its objectives, and the software development methodology used during its elaboration. In the end, we present the document structure with the different sections within it and a brief description of them.

1.1. Motivation

Communication is a pillar in interpersonal relationships, a fundamental need in our society. Unlike most people, for some, this action requires a lot of effort. Their differences create an uncrossable barrier and almost impossible human connection using the traditional way of communication. To remove this barrier an alternative approach should be used - pictograms. These graphic images, representing an object or a concept, have helped a lot to establish an initial communication channel between people with special needs and the rest. Although there are specialists trained to work and teach this unique language, for the majority of people, these pictograms are unknown and their purpose is not fulfilled.

To completely include pictograms and transform the communication between people with disabilities and those without, it is needed to harvest the capabilities of modern technologies, and create software. Software, which can “translate” pictograms into natural language - Spanish.

Currently, multiple tools allow transforming natural language into pictograms. The initial version of “Pict2Text” goes a step further and does the reverse action - translating the pictograms into natural language. But to realize this, it requires a sentence to be written using pictograms that are selected manually searching for every one of them. This is not good enough as the people with disabilities who need the pictograms, cannot write and search the specific word so that it can be matched to the pictogram they want. This

problem can be solved, giving those people the option to upload a picture of a sentence constructed with pictograms. With the next generation of “Pict2Text”, version 2, we aim to reduce the exclusion of those people with special needs from society and break the communication barrier, providing the aforementioned functionality.

The beneficiaries of this software will be the above mentioned two groups, people with communication problems and those without them. In the first group, we include individuals who use pictograms to communicate, those with cerebral palsy, autism, or any other kind of cognitive disability. The second group is for all the others, users of the natural language. This project is created to help both of the previously mentioned groups, integrating the first one as a normal part of the society, able to communicate freely, expressing their needs, feelings, desires, and the second not only as assistants for the needs of the first group but also as friends and equal. In this modern, 21st century world, we all should be equal independently of if we can or cannot use the natural language in our communication.

1.2. Goals

The main goal of this project is the creation of an image classification machine learning model capable of distinguishing single or multiple pictograms from a given image.

Integrate a pictogram classification model as new functionality and extend the previous version of “Pict2Text”.

Use services-oriented application architecture, constructing web services, and/or microservices.

Use industry standards during the whole process.

Consolidate and expand the knowledge acquired during the period of acquiring the Software Engineering Degree.

1.3. Software development methodology

Given our initial inexperience with the problem we wanted to solve, we decided to use an agile methodology. Unlike the traditional methodologies, that are more suitable when you have clearly defined requirements, and follow a linear approach - initiation, planning, execution, monitoring, and closure, agile methodologies are more flexible and focus less on initial planning. The small iterations will allow us to receive feedback before the end of the final project, which makes it easier to correct mistakes on time.

After further consideration, we selected Kanban as the most suitable methodology for us. The product is delivered continuously, and changes are allowed during the whole process, and no estimation of the tasks is needed. That will increase our flexibility and productivity and will decrease the effect of our lack of experience in estimating.

During every meeting, the tutors will give us the tasks that we have to finish until the next one, including their priority. The date for the next meeting also will be decided during the current one, as the sprint may vary between two and four weeks, depending on the given tasks. We decided to have the following columns:

- To do. Here we will put the tasks given to us by the tutors in the order they determine.
- In progress. This column contains the tasks we are currently working on. All tasks in that column will have a particular person assigned. As we are using Kanban, we have limited the WIP (Work In Progress) of the column to 4 assignments to avoid doing more tasks than we can reasonably manage. If more tasks come from the column "Testing" or "Ready for review", increasing the number of tasks more than the given WIP, some of them will be moved to the column "On hold".
- On hold. Here we will put the activities that we can't continue at the moment. The reason behind this could be that they are waiting for another task to finish. When the task could be continued, it is, once again, moved to "In progress".
- Testing. In the case of programming assignments, the testing will include code review and automated and/or manual testing. If documentation is in this column, the person who didn't write the particular part will read and correct it. When the testing is finished, the task is moved to "In progress" if a bug is spotted or the task is not finished, or to "Ready for review" in the opposite case.
- Ready for review. This column contains the tasks we have finished but are not yet reviewed by our tutors. The tasks in that column will be reviewed during the next meeting and the tutors will decide which of them will be moved to the column "Done" and, in case of a task that is not completed, to the column "In progress".
- Done. The last column contains the tasks that are finished, reviewed, and approved by the tutors.

1.4. Document structure

The document is structured as follows.

Chapter 1(Introduction) where we have covered the motivation, the goals, and the methodology for the project.

Chapter 2(State of the Art) describes pictograms and existing tools for their translation, covers the state of the first version of “Pict2Text”, the functionalities we will include to increase its usefulness, and the general idea of the machine learning algorithms and models we will be using.