



Co-funded by the  
Erasmus+ Programme  
of the European Union

**FOSTERING AI AT SCHOOLS**  
<http://fosteringai.net>

# BRAGA LTTA EVENT - FAIAS

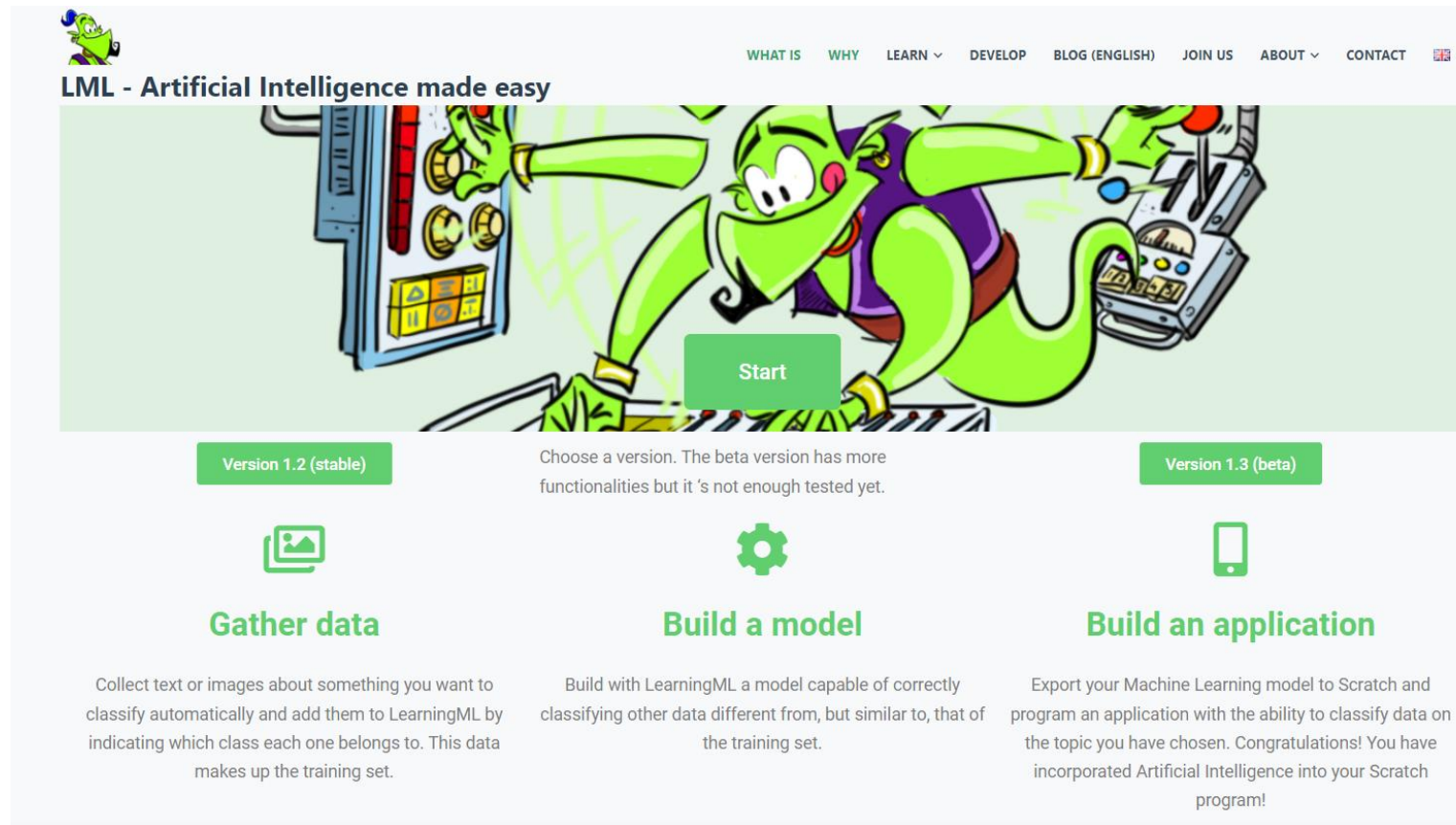
LEARNINGML ACTIVITY

Image & Text Recognition

01/06/2022

by Antonio José Romero Barrera

## LEARNINGML WEB

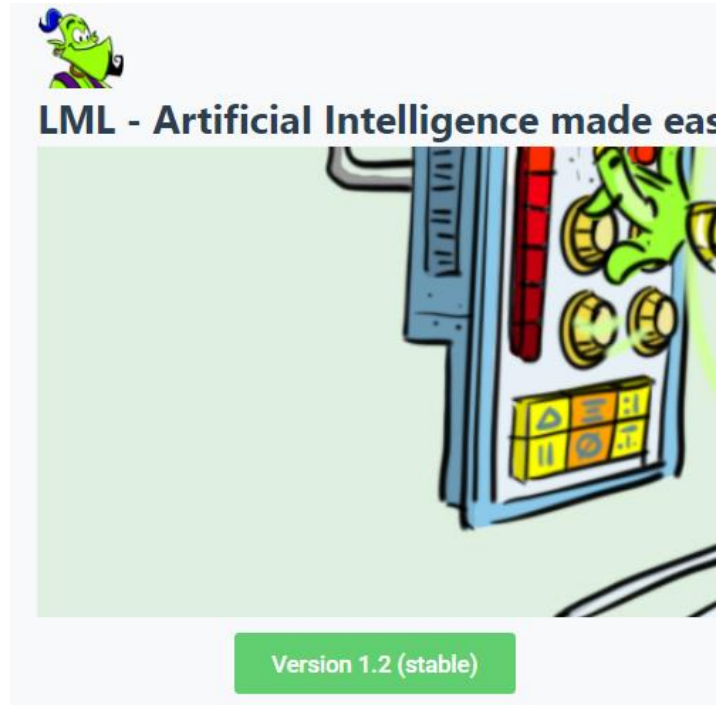


The screenshot shows the LearningML website interface. At the top left is a small cartoon character logo. To its right is a navigation menu with links: WHAT IS, WHY, LEARN (with a dropdown arrow), DEVELOP, BLOG (ENGLISH), JOIN US, ABOUT (with a dropdown arrow), and CONTACT (with a flag icon and a dropdown arrow). Below the navigation is the main heading "LML - Artificial Intelligence made easy". Underneath is a large illustration of a green cartoon character with a purple vest, interacting with a control panel and a device. A green "Start" button is overlaid on the illustration. Below the illustration are three columns of content:

- Version 1.2 (stable)**: A green button with a folder icon. Below it, the text "Gather data" is followed by a paragraph: "Collect text or images about something you want to classify automatically and add them to LearningML by indicating which class each one belongs to. This data makes up the training set."
- Version 1.3 (beta)**: A green button with a gear icon. Below it, the text "Build a model" is followed by a paragraph: "Build with LearningML a model capable of correctly classifying other data different from, but similar to, that of the training set."
- Version 1.3 (beta)**: A green button with a smartphone icon. Below it, the text "Build an application" is followed by a paragraph: "Export your Machine Learning model to Scratch and program an application with the ability to classify data on the topic you have chosen. Congratulations! You have incorporated Artificial Intelligence into your Scratch program!"

<https://web.learningml.org/>

## LEARNINGML WEB



**LearningML** consists of two javascript-based applications:

- The Machine Learning model editor (learningml-editor): It is the tool with which ML classification models are created by collecting labeled examples.
- The lml-scratch programming editor: lml-scratch is a modification of the well-known Scratch project. That is to say, I have taken the original Scratch code and added the necessary code to communicate with the ML model editor and to incorporate new blocks.

### Stable version 1.2:

This is the default version of the webpage, it has two functionalities already tested and with optimal operation (*Image recognition* and *Text recognition*)

<https://web.learningml.org/>

## LEARNINGML WEB

9

español

english

galego

catalá

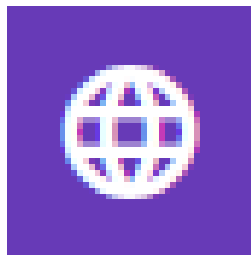
euskera

italiano

deutsch

Ελληνικά

português



7

español

english

galego

català

italiano

deutsch

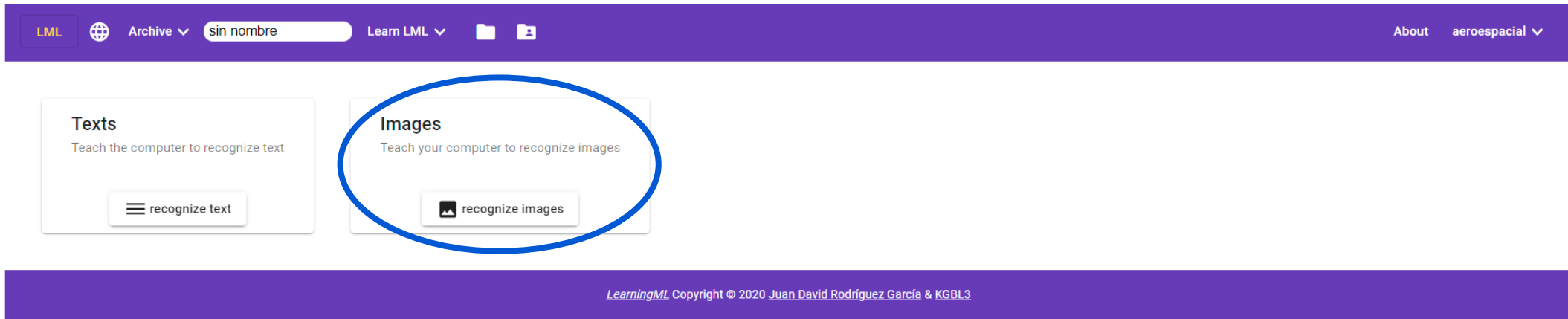
Ελληνικά

AVAILABLE LANGUAGES IN BETA VERSION

AVAILABLE LANGUAGES IN STABLE VERSION

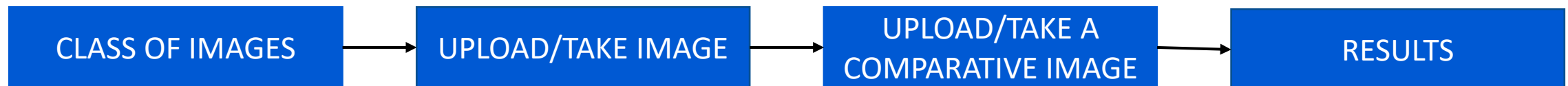
<https://web.learningml.org/>

# PART 1: IMAGE RECOGNITION



The screenshot shows the LearningML web interface. The top navigation bar is purple and contains the LML logo, a globe icon, an 'Archive' dropdown menu, a search bar with 'sin nombre' entered, a 'Learn LML' dropdown menu, and icons for folders and a document. On the right side of the navigation bar are 'About' and 'aeroespacial' dropdown menus. Below the navigation bar, there are two main sections: 'Texts' and 'Images'. The 'Texts' section has the subtext 'Teach the computer to recognize text' and a button labeled 'recognize text'. The 'Images' section has the subtext 'Teach your computer to recognize images' and a button labeled 'recognize images'. The 'Images' section and its button are circled in blue. At the bottom of the page, there is a purple footer bar with the text: 'LearningML Copyright © 2020 Juan David Rodríguez García & KGBL3'.

**IMAGES:** ABILITY OF THE SOFTWARE TO IDENTIFY OBJECTS AND GEOMETRIES TO ELABORATE A SERIES OF RELATIONS BETWEEN DIFFERENT IMAGES IN ORDER TO ASSOCIATE THEM TO ONE ESTABLISHED GROUP



# PART 1: IMAGE RECOGNITION

## LEARNINGML IMAGE EDITOR INTERFACE


### 1. Train

First I need some image examples

 Add new class of images

### 2. Learn

Now it's time to learn to classify images

 Learning to recognize images

### 3. Try

Introduces new terms and checks they are correctly classified



## PART 1: IMAGE RECOGNITION



**ADD NEW CLASS OF IMAGES:** create a new set add images the related examples



**ADD IMAGES:** import to the created class/comparative image a new image from your computer



**ADD IMAGES FROM THE WEBCAM:** use your hardware to take photos and import them into the created class or your comparative image



**ADD NEW CLASS OF IMAGES:** create a new set add images the related examples



**REMOVE TAG:** delete a created class



**SCRATCH PROGRAMMING:** write a program in Scratch which is able to recognise similar but different images to the ones that you used to teach the computer

# PART 1: IMAGE RECOGNITION

## STEP 1. TRAINING

- Collect examples of texts you want to recognise.
- You can add new tags or classes (= names of the types of things you want to recognise with LML).
- A minimum of 10 examples of each class are recommended.

## STEP 2. LEARNING

- In the default mode, it's a black box method to teach the computer to understand your examples added in the training part to recognise new images and associate them with your classes.
- In the advance mode you can edit some parameters of the process.

## STEP 3. TRY

- Import your input to compare with the examples of the different classes created previously.



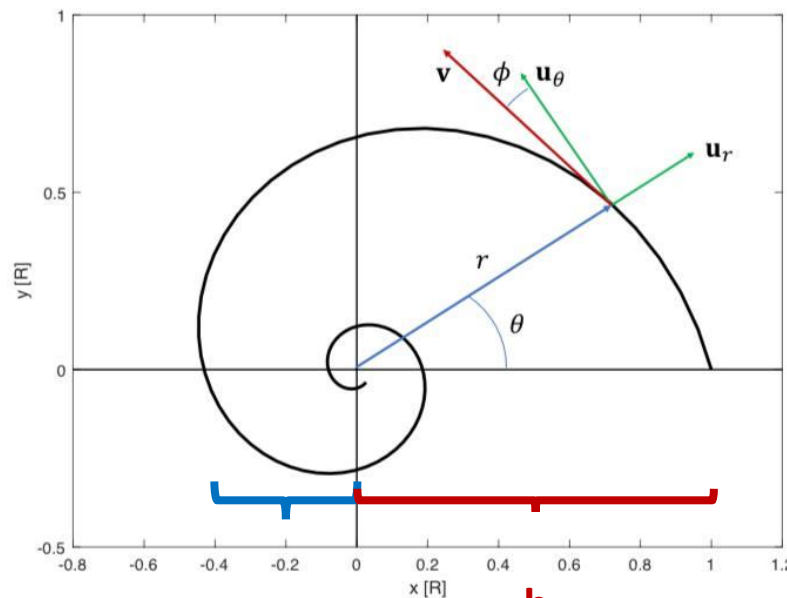
# PART 1: IMAGE RECOGNITION

## EXAMPLE PART 1: THE 5 PATTERNS/SERIES IN NATURE (MATH LESSON)

1. **Identify** the class names (what's the patterns names and number of it?) → 5 classes.
  - Spiral pattern, Fractal pattern, Symmetry pattern, Dunes/waves pattern & Chaos pattern.
2. **Create** the different classes.
3. **Upload** the corresponding images to each class (10 images recommended for each class).
4. **Train** the AI to detect coincidences between the images of each group.
5. **Try** the model uploading a new image input to compare.

# PART 1: IMAGE RECOGNITION

## EXAMPLE PART 1: THE 5 PATTERNS/SERIES IN NATURE (MATH LESSON)



$$\Phi = a/b = (a+b)/a$$

**a**

**b**

1.6180339887...

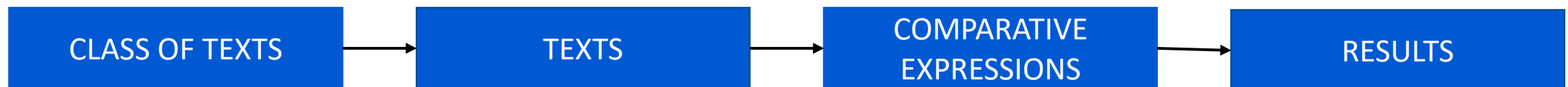
<https://learningml.org/editor/model/image>

## PART 2: TEXT RECOGNITION



The screenshot shows the LearningML web interface. At the top, there is a navigation bar with 'LML', a globe icon, 'Archive', a search bar containing 'sin nombre', 'Learn LML', and icons for folders and documents. On the right, there are links for 'About' and 'aeroespacial'. Below the navigation bar, there are two main sections: 'Texts' and 'Images'. The 'Texts' section is circled in blue and contains the text 'Teach the computer to recognize text' and a button labeled 'recognize text'. The 'Images' section contains the text 'Teach your computer to recognize images' and a button labeled 'recognize images'. At the bottom of the interface, there is a footer with the text 'LearningML Copyright © 2020 Juan David Rodríguez García & KGBL3'.

**TEXTS:** EXTRACTION, ASSOCIATION AND ANALYSIS OF THE INFORMATION AND STRUCTURE OF TEXT INPUT IN ORDER TO DETERMINATE ITS CLASS MEMBERSHIP




## PART 2: TEXT RECOGNITION

### LEARNINGML TEXT EDITOR INTERFACE

#### 1. Train


First I need some text examples

 Add new class of texts

#### 2. Learn

Now it's time to learn to classify text

Language of texts

 Learning to recognize text

#### 3. Try

Introduces new terms and checks they are correctly classified

Expression

---

Check



## PART 2: TEXT RECOGNITION

### TRAINING

- Collect examples of texts you want to recognise.
- You can add new tags or classes (= names of the types of things you want to recognise with LML).
- A minimum of 10 examples of each class are recommended.

### LEARNING

- In the default mode, it's a black box method to teach the computer to understand your examples added in the training part to recognise new texts and associate them with your classes.
- You need to select the language of you
- In the advance mode you can edit some parameters of the process.

### TRY

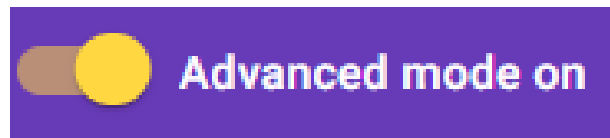
- Write your input to compare with the examples of the different classes created previously.

## PART 2: TEXT RECOGNITION

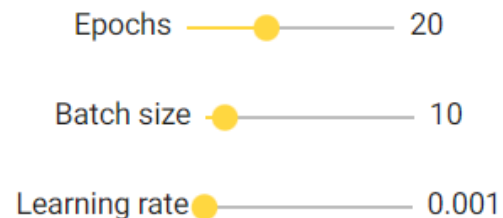
### EXAMPLE PART 2: 4 GEOGRAPHICAL ACCIDENTS (GEOLOGY LESSON)

1. **Identify** the class names (what's the patterns names and number of it?) → 4 classes.
  - River, Mountain Range, Plains, Depression.
2. **Create** the different classes.
3. **Upload** the corresponding images to each class (10 text examples recommended for each class).
4. **Train** the AI to detect coincidences between the texts of each group.
5. **Try** the model writing a new text input to compare.

## ADVANCED MODE (AVAILABLE IN V1.3)



This mode offers us some control over the execution of the algorithm, as well as information about the result.



- Adjustable parameter of the Neuronal Network:
  - **Batch Size:** defines the number of samples to work through before updating the internal model parameters.
  - **Epochs:** defines the number times that the learning algorithm will work through the entire training dataset.
  - **Learning rate:** determines the step size at each iteration while moving toward a minimum of a loss function.