

Trabajo final: detección de señales

Visión artificial



9 de diciembre de 2020

David rodríguez y Alejandro Meza

**Índice**

**1.Tecnologías usadas**

**2.Metodología**

**3.Planificación**

**4.Objetivo 1**

**5.Objetivo 2**

**1.Tecnologías usadas**

Con el fin de desarrollar el trabajo de una manera más cómoda, se ha utilizado GitHub para gestionar las versiones del proyecto y poder almacenar archivos necesarios para los objetivos.

**2.Metodología**

La metodología de trabajo que se ha seguido ha sido iterativa e incremental; produciendo una versión nueva del programa en cada iteración.

**3.Planificación del proyecto**

*Planificación de objetivo 1*

|  |  |  |  |
| --- | --- | --- | --- |
| **Autor de la versión** | **Número de la versión** | **Fecha** | **Descripción** |
| Todos | Versión 1.0 | 02/12/2020 | Lectura de imágenes de las carpetas de los datasets de manera exitosa; previamente habiendo aplicado un resize. |
| Todos | Versión 1.1 | 03/12/2020 | Obtención de descriptores, mediante diversas técnicas. |
| Todos | Versión 1.2 | 08/12/2020 | Corrección de aspectos fallidos de la versión anterior y creación de diccionario. |
| Todos | Versión 1.3 | 09/12/2020 | Implementación de funciones para obtener descriptores sobre cierto grupo de imágenes mediante diversas técnicas. |
| Todos | Versión 1.4 | 26/12/2020 | Creación del dataset e implementación básica de regresión logísitica. |
| Todos | Versión 1.5 | 27/12/2020 | Implementación de gráficas en el modelo de regresión logística. |
| Todos  -------------  Todos | Versión 1.6  --------------  Versión final | 28/12/2020  ----------------  29/12/2020 | Corrección de aspectos fallidos de la versión anterior.  -------------------------------------------------Implementación de SVM. Obtención de resultados definitivos en SVM y regresión logística. Elección definitiva de mejor modelo |

**4. Objetivo 1**

Para cumplir el objetivo 1, se ha dividido este en dos tareas distintas:

* Creación de datos: a partir de las imágenes, se han obtenido sus descriptores; siendo estas las características que hemos usado.
* Utilización de modelo de predicción: A partir del conjunto de características que ha sido obtenido, se usa un modelo existente para predecir nuevos datos.

1. Creación de datos

1.1 Relacionar carpetas con imágenes

Para la creación de datos, se ha creado primero una función que sea capaz de relacionar el nombre de cada carpeta de señales con el tipo de señales que contiene. Se consideró que hacer esto es fundamental, pues aportará mayor facilidad después a la hora de clasificar imágenes y una visión más intuitiva del código.

Después, se creó una función, que, con ayuda de la anterior, es capaz de crear un diccionario cuyas claves son el tipo de señal, y sus valores son las imágenes.

* 1. Obtención de descriptores

Tal como se puede apreciar en el notebook, se han realizado una serie de funciones capaces de obtener descriptores mediante diferentes técnicas*: HOG, SHIFT, FASTBRIEF y ORB*.

El objetivo principal de un descriptor de características es poder generalizar el objeto de tal manera, que el mismo objeto produzca un descriptor de características similares cuando se vea afectado por diferentes condiciones.

Pese a la idea inicial de probar con distintos métodos, los que no son HOG devuelven descriptores de dimensiones no uniformes; luego para crear un dataset, esto resulta verdaderamente problemático.

Por tanto, la técnica elegida ha sido HOG. HOG significa simplemente “histograma de gradientes orientados”. Se centra en contar la ocurrencia de la orientación del gradiente en partes concretas de una imagen.

* 1. Creación de dataset

Tras obtener las características, se ha creado el dataset con el que se va a trabajar. Se obtienen el conjunto de entrenamiento y de prueba. El porcentaje ha sido del 60% y 40% respectivamente.

1. Utilización de modelo de predicción

Una vez con los conjuntos de dato creados, es momento de aplicar modelos de *machine learning para* conseguir clasificar los distintos tipos de señales que hay.

Si bien es cierto que se pudo realizar un único modelo, se consideró positiva la idea de probar dos modelos distintos y enfrentarlos entre sí. Decidiendo a partir de la precisión en el conjunto de prueba, cuál de los dos modelos es el ideal.

En primer lugar, se probó **regresión logística**. Debido a que si 𝑛 es grande (relativo a m), siendo n el número de características y m el número de ejemplos se recomienda usar regresión logística. Como ese no es el caso exacto de nuestro problema, se pensó que no se iba a obtener un buen resultado. Efectivamente, eso ocurrió.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

El porcentaje de acierto en el conjunto de prueba **no llegó ni al 15%;** por lo que se descartó inmediatamente el modelo.

Como alternativa, se pensó en la utilización de **SVM**, basándose en algunos artículos que lo señalaban como un modelo competente para la clasificación de imágenes.

Al crear el modelo se aplicó *GridSearchCV*. ¿En qué beneficia su uso? Un clasificador SVM tiene una serie de hiperparámetros; como lo son C o gamma. Encontrar una combinación de estos que proporcione un funcionamiento ideal, puede ser una tarea bastante compleja, así que puede considerarse crear una cuadrícula -*grid*- y probar todas las combinaciones posibles. Es ahí donde *GridSearchCV* entra en escena, y elige la mejor combinación de manera automática.

Tras probar el modelo con los datos de entrenamiento, se consiguió un resultado que superaba a la regresión logística de sobremanera: una **precisión del 83%.** Por tanto, de cara al objetivo número dos se consideró exclusivamente SVM.

Imagen que contiene Tabla

Descripción generada automáticamente