**MATERIA**

Proyecto Integrador

**TÍTULO**

Innovación Tecnológica para la Implementación de ML en la Industria de Manufactura: con Enfoque en Pattern Analyzer y Cosmetic Vision System

**Avance 2. Ingeniería de características**

**INTEGRANTES – EQUIPO 35**

A01793810 - Jerson David Pérez Contreras

A01228278 - Norma de los Ángeles García López

A01794256 - Angel De Jesús Hernández Pascual

**FECHA**

12/Mayo/2024

A. Se aplicarán operaciones comunes para convertir los datos crudos del mundo real, en un conjunto de variables útiles para el aprendizaje automático.

Las imágenes fueron normalizadas y preprocesadas con la aplicación que se desarrolló para la implementación en Amatek.


Imagen 1.- Imagen principal de la aplicación vision System

A continuación se expone el código que se utiliza para realizar la eliminación de ruido en la imagen:

***Image<Bgr,byte> SnapImage = Image.ToImage<Bgr, byte>();*** *//Se convierte el bitmap a formato Image<Bgr,byte> de OpenCV la cual servirá como base a la cual se le aplicará el filtro*

***Image<Hsv, byte> Snap = Image.ToImage<Hsv, byte>();*** *//Se convierte el bitmap a formato Image<Hsv,byte> la cual se le aplicará el filtro para extraer el color ámbar*

***Snap = Snap.SmoothMedian(7);*** *//Se aplica el algoritmo SmoothMedian a la imagen HSV para reducir el ruido, principalmente en la zonas de color obscuro.*

***ColorE filter = ColorE.GetLimits(ColorType.ambar);*** *//Se obtienen los valores Low y High Del HSV*

***var Mask = Snap.InRange(filter.Low, filter.High);*** *//Se crea la maskara (Imagen 2)*

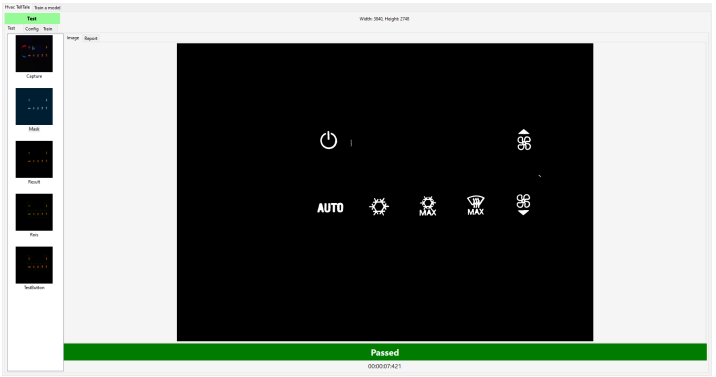


Imagen 2.- Ejemplo de la imagen de Mask

***Mask = Mask.Erode(1);*** *//Se aplica la función Erode para eliminar los bordes*

***Mask = Mask.Dilate(1);*** *//Se aplica la función Dilate, para eliminar bordes sin reducir el tamaño de los objetos*

***Mask = VisionClass.FindContours(ref Mask, 100, true);*** *//Se eliminan las partículas menores a 100 pixeles*

***return Mask.Convert<Bgr, byte>().And(SnapImage).ToBitmap();*** *//Se aplica la máscara a la imagen original dejando solo los pixeles en el rango de ámbar. (imagen 3)*

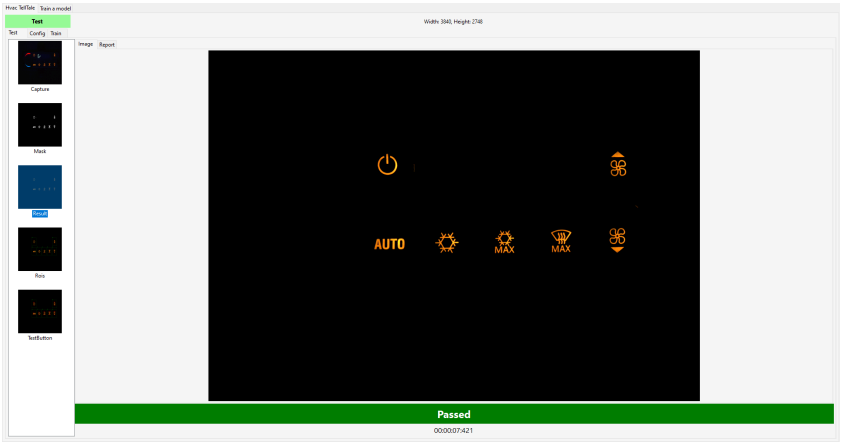


Imagen 3.- Ejemplo de la imagen de Mask

Con el objetivo de eliminar el ruido generado por la iluminación y resaltar la importancia de las formas de los iconos y el color, se decidió aplicar el algoritmo anterior dando como resultado:

Imagen 4. Imagen sin procesar

Imagen 5. Imagen procesada

Como se puede observar en la imagen 5 se resalta únicamente los píxeles del icono, eliminando completamente el ruido.

Uno de los principales problemas al generar estas imágenes es que el logo está iluminado por un led detrás de una membrana transparente difusa, lo cual crea mucha variabilidad en la tonalidad del led y la intensidad de los píxeles.

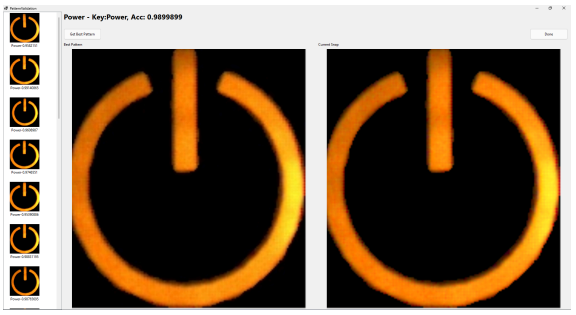


Imagen 6.- En caso de solo elegir un método por Pattern Match nuestra herramienta tiene la capacidad de proveernos nuestro mejor Pattern de su biblioteca, procesandolos a través de de ella y extrayendo el pattern con el mejor Accuracy.

# B.- Se analizan los resultados de nuestro entrenamiento.

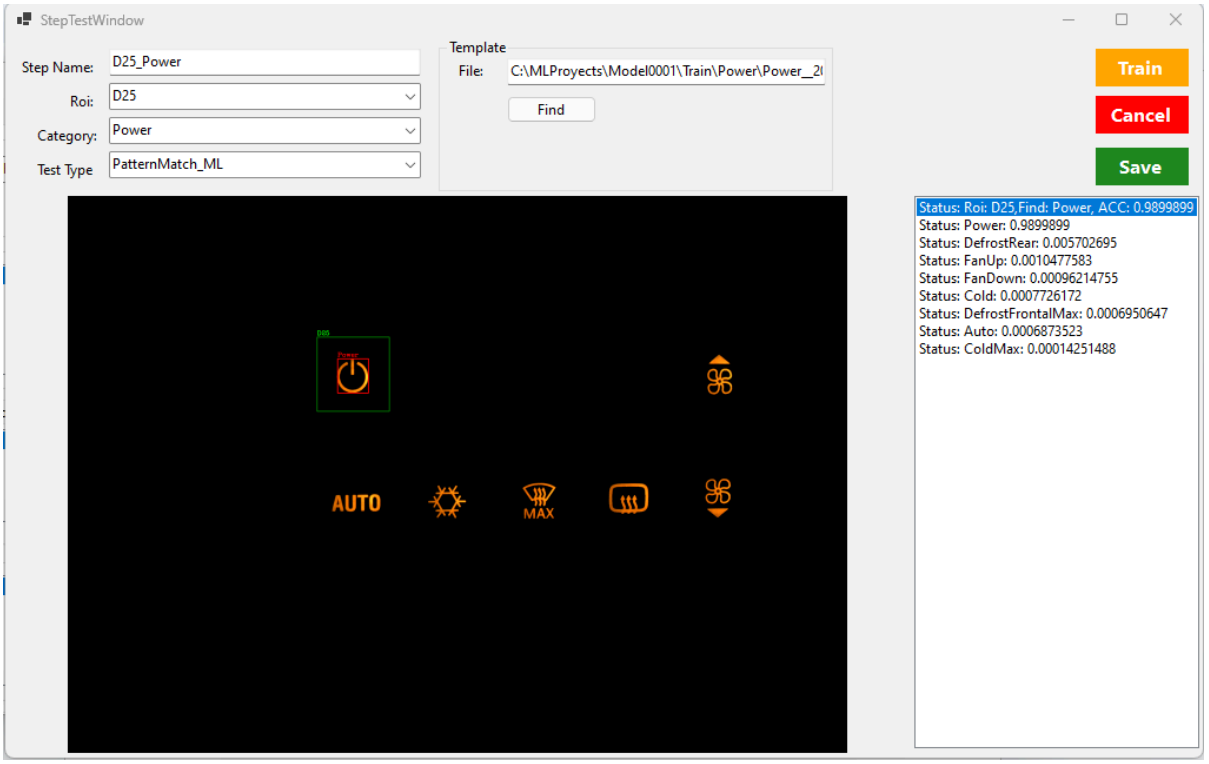


Imagen 7.- Una vez seleccionada la categoría nuestro sistema identificará el ROI requerido y procesara la imagen para obtener la clasificación a través de nuestra ML .

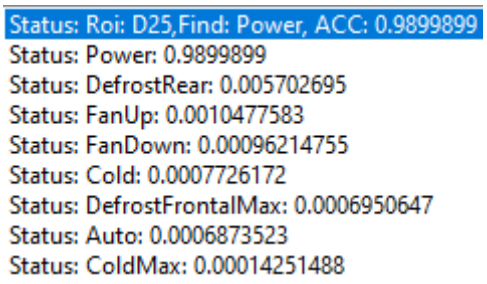


Imagen 8.- El resultado del procesamiento es el anterior siendo la clase Power la más cercana a 1 con un valor de accuracy de 0.9899.

El código de la aplicación se encuentra dividido en 4 proyectos que ya se encuentran en GitHub:

https://github.com/AngeHdz/VisionSystemConfigFile.git

https://github.com/AngeHdz/VisionSystemAmetek.git

https://github.com/AngeHdz/MLClass.git

https://github.com/AngeHdz/EmguClass.git

# C.- Conclusiones

El enfoque principal de la preparación de nuestras imágenes es resaltar la forma y el color de nuestro iconos, mediante el uso del enmascaramiento mediante un rango HSV, el cual es el más eficientes a la hora de crear filtros.
Se implementaron filtros SmoothMedian de 7x7, y el algoritmo Erode y Dilate. Se eliminaron los objetos menos a 100 pixeles.
Una vez enmascarado los iconos generan datos muy cercanos a 1 como resultado accuracy según los datos de nuestras pruebas.

La ventaja del algoritmo de ML.net que implementa el modelo ML y OpenCV comparado con el proceso que actualmente se utiliza de National Instrument está limitado a máximo 15 clasificaciones y el nuevo algoritmo no tiene una capacidad máxima , esto sin duda es una gran ventaja competitiva que ayudará a tener una mejor estándar de calidad ya que este algoritmo será un aliado para los operarios al momento de ensamblar y por ende disminuir los desperdicios de materiales.