

Detección del Nivel de Manómetro Analógico por Imágenes

-El repositorio del trabajo se encuentra en el siguiente link:

https://github.com/k1ch4n/MCC616_InformeFinal_G6

Llallihuaman Calderon, Cristian Marko

Estudiante Maestría Ciencias de la Computación (FC-UNI)

email: cllallihuamanc@uni.pe

Arias Espada, Roberto Jesus

Estudiante Maestría Ciencias de la Computación (FC-UNI)

email rariase@uni.pe

Resumen—Mediante el presente proyecto el sistema mediante imágenes tomadas del manómetro detecta si la presión se encuentra en los niveles permitidos, para el desarrollo se uso la biblioteca de machine learning Tensorflow en el entorno Python.

Index Terms—manometro, imagenes, Tensorflow, Machine Learning

I. INTRODUCCIÓN

La mayoría de los equipos utilizados en la industria utilizan el control de presión debido al transporte de vapor o líquidos de los insumos utilizados en sus procesos industriales.

Por este motivo es necesario siempre tener el control a través de los medidores de presión denominados manómetros, hay de dos tipos el digital y el analógico siendo este ultimo el mas utilizado en la industria por la reducción de costo y el tiempo en realizar mantenimiento en comparación con al digital, para verificar la medida que marca el manómetro se complica en algunas situaciones ya sea por el tiempo en verificarlo constantemente o por la zona donde esta situada por ser difícil de acceder por condiciones de temperatura o accesibilidad.

El objetivo de este proyecto es otorgar a un sistema de alerta el aviso cuando las condiciones de presión no son las adecuadas mediante imágenes tomadas del manómetro que serian captadas en forma periódica para controlarlo sin necesidad de un operario.

II. ESTADO DEL ARTE

Se revisaron publicaciones y tesis que procesan las imágenes de manómetros y relojes. Para la realización del código se reviso la bibliografía de la biblioteca del Keras.

III. INSTRUMENTACIÓN A USAR

El manómetro analógico tipo Bourdon esta calibrado para medir de 0 a 4 bar como se aprecia en la Figura 1 , se capturo el vídeo usando una cámara de 8 megapixeles de un celular Motorola G8 y para el desarrollo computacional se trabajo en lenguaje Python usando la biblioteca Tensorflow.



Figura 1. Manómetro usado en el proyecto

IV. METODOLOGÍA

IV-A. Entrada de datos

Se capturo de un vídeo de unos minutos del manómetro como varia su presión en su proceso de funcionamiento del sistema para ser separadas en frames, con las imágenes obtenidas se etiquetara en las clases de bajo y normal indicando el valor bajo para cuando el sistema esta por debajo de la presión necesitada (0.5 bar) y normal cuando la presión esta en los niveles requeridos.

IV-B. Trabajo de datos

Se realizó el pre procesamiento de las imágenes usando las librerías matplotlib, PIL, numpy y Keras del Python, las imágenes están estructuradas en dos carpetas: una carpeta “entrenamiento” donde se encuentra las imágenes del entrenamiento y “test” donde se encuentran las imágenes para el entrenamiento, así cada una de las carpetas están divididas en dos carpetas más con los estados óptimos (“normal”) y los estados donde debe enviar la alerta (“bajo”), separadas en

carpetas el Keras inicia el preprocesamiento para preparar la dataset dando como salida imágenes de (64,64) .

Para el modelo se utilizan los comandos Conv2D, Max-Pooling2D, Flatten, Dense de la siguiente manera: primero se realiza una convoluciones con una activación ReLu con un total de 32 filtros para reducirlos a la mitad, luego de repetir el proceso una vez más se aplanan en un array.

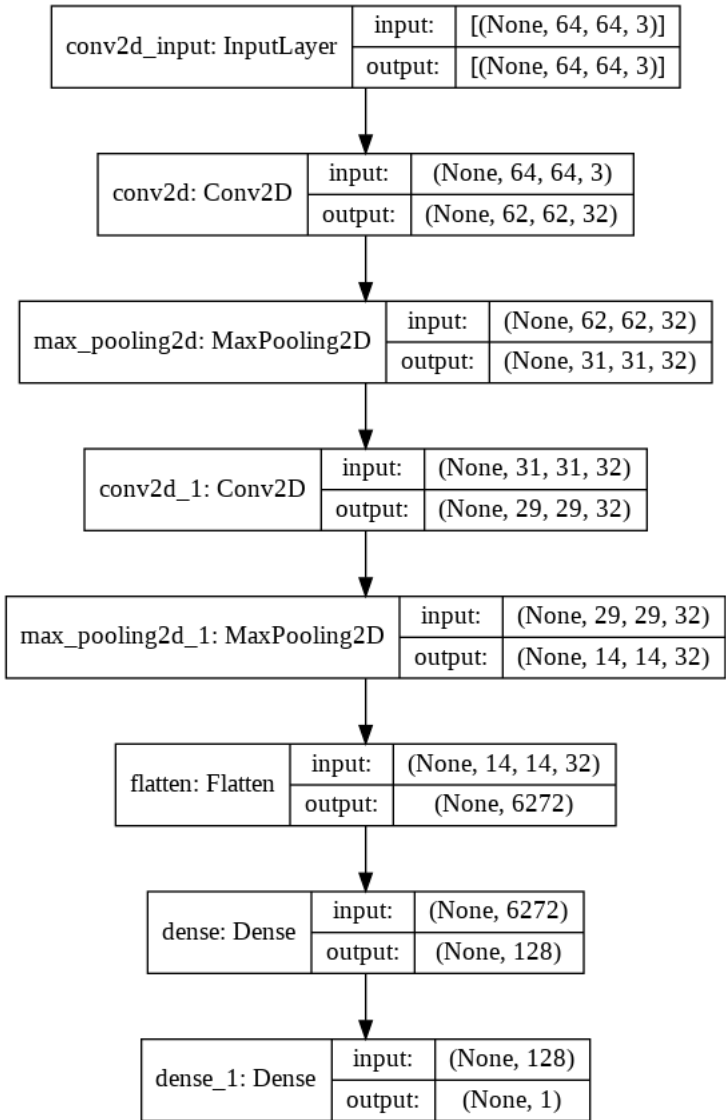


Figura 2. Diagrama del modelo

Para terminar se crean 128 neuronas con activación ReLu y la siguiente con sigmoide dando un 0.87 % de predicción , por recomendación se añadió de las pruebas una última capa sigmoide para aumentar la predicción, en la compilación se utilizo el Adam clasificando para dos salidas, el modelo queda preparado como se muestra en la 3.

Con los datos etiquetados se procede a prepararse para el entrenamiento convirtiéndolo en un dataframe.

Model: "model_2"

Layer (type)	Output Shape	Param #
input_3 (InputLayer)	[(None, 2048)]	0
batch_normalization_v2 (Batch Normalization)	(None, 2048)	8192
dense (Dense)	(None, 256)	524544
dense_1 (Dense)	(None, 256)	65792
dense_2 (Dense)	(None, 2)	514
Total params: 599,042		
Trainable params: 594,946		
Non-trainable params: 4,096		

Figura 3. Manómetro usado en el proyecto

IV-C. Ajustar la CNN a las Imágenes

Para el ajuste de imágenes se importa de la librería Keras el ImageDataGenerator realizando un cambio de escala para que los valores estén entre bajo y normal tanto para las imágenes de entrenamiento y evaluación, en las imágenes de entrenamiento se aplica una intensidad de corte (ángulo de corte en sentido antihorario en grados) de 0.2 y volteando las entradas en forma aleatoria horizontalmente con 10 rondas.

En la generación de datos se usa de la librería Keras el flow_from_directory dimensionando las imágenes a 64x64 etiquetándolo en forma binaria.

Para procesar las imágenes de entrenamiento de 10 pasos por épocas siendo 5 veces que el generador del conjunto de validación devuelve.

V. RESULTADOS

Para 10 épocas se obtienen los siguientes resultados figura 5.

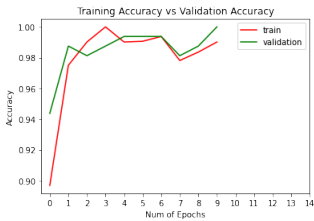


Figura 4. Curvas Entrenamiento y Validación

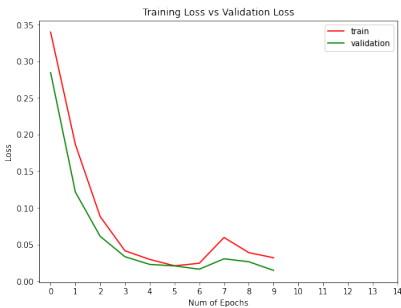


Figura 5. Curvas de Perdidas por Entrenamiento y Validación

VI. CONCLUSIONES

- La predicción mejora con el numero de épocas haciendo valido el uso en manómetros analógicos.
- Una mejora de forma de trabajar el problema de clasificación es en forma de regresión para acceder de forma precisa el nivel de presión.

REFERENCIAS

- [1] Alvarelllos, Aa, Picón, D.b , Puertas, J.b, Rabuñal, Jc, “Sistema para medición de manómetros analógicos mediante visión artificial”.
- [2] Jorge Eduardo Linares Restrepo, Lauren juliana Galvis Aristizabal “CONTROL DE PRESIÓN USANDO VISION ARTIFICIAL A TRAVÉS DE UN Procesador Digital de Señales DSP”.