UNIVERSIDAD SEÑOR DE SIPÁN

UNIVERSIDAD SEÑOR DE SIPÁN

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO
ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS

TEMA DE TESIS

IDENTIFICACIÓN AUTOMÁTICA DE CARACTERES NUMÉRICOS EN IMÁGENES DIGITALES DE CONSUMO DE ENERGÍA ELÉCTRICA.

PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS

AUTOR:
Bach. Kevin Alehexis Guevara Cabrera

Línea de la Investigación:
Ciencias de la Computación

Pimentel – Perú
2017

IDENTIFICACIÓN AUTOMÁTICA DE CARACTERES NUMÉRICOS EN IMÁGENES DIGITALES DE CONSUMO DE ENERGÍA ELÉCTRICA

APROBACIÓN DE TESIS

Dr. Gutierrez Gutierrez Jorge Luis
Presidente de jurado de tesis

Mg. Tuesta Monteza Victor Alexci
Secretario(a) de jurado de tesis

Ing. Ocampo Moreno Liliana
Vocal de jurado de tesis

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

DEDICATORIA

Agradecer a Dios por mi madre, al brindarme su apoyo desde que inicie la universidad y su preocupación por mi salud; a mis hermanas en comprenderme y responder a mis interrogantes en torno a mis acontecimientos en el aspecto personal, además por rodearme de personas excelentes que me apoyaron en mi vida personal y profesional.

AGRADECIMIENTO

A Dios por permitirme a desarrollar esta investigación y conocer en el camino a profesionales para orientarme en el tema de mi investigación; me refiero al Ingeniero Jonatan Sanchez Hinostroza, que me incentivo con la idea de esta investigación; además al Ingeniero Joel Vásquez Villalobos, por indicarme las pautas para el desarrollo en la investigación; y al Ingeniero Manuel Guillermo Forero Vargas, de la ciudad de Colombia, por la idea del protocolo de capturas y las pautas para su desarrollo.

Índice

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Resumen

Los medidores de consumo de agua, luz y gas son instrumentos para medir la cantidad de consumo del cliente en el servicio suministrado por las empresas de servicios ya sean públicas o privadas. Actualmente algunas empresas siguen gestionando la información del servicio prestado en hojas manualmente o haciendo uso de aplicaciones donde se digitan los datos, realizar esta tarea de esta manera genera errores y gatos de tiempo para recolectar y verificar los datos ingresado.Para realizar este cambio seria la realización de un reconocimiento óptico de caracteres (OCR) que se puede ser implantados en unos celulares inteligentes y así agilizar el proceso de capturas y validación de los datos.

Hoy los celulares inteligentes han avanzado con gran progreso que cada temporada saca nuevos equipos con nuevas estructuras y la calidad de las imágenes con buenas resoluciones.

Se utilizó 1300 imágenes de consumo de medidores de energía eléctrica como bases de datos y alrededor de 3000 imágenes de caracteres extraídos por las imágenes originales, también se planteó un protocolo de tomas fotográficas desde una aplicación móvil y la utilización de librería libres como OpenCV.

Para el entrenamiento se realizo es hacer el entrenamiento de las 3000 imágenes de los caracteres extraídos y convertirlos en un archivo para luego en la clasificación reutilizarlo.

Y posteriormente ejecutar con un medidor y el reconocimiento de los caracteres con la utilización del clasificador KNN con el más óptimo para la clasificación ya que sus resultados en la probabilidad es el 40 al 60% y basado al reconocimiento basado a la precisión en el 80% y con una red neuronal ha tenido una precisión del 72.7%.

PALABRAS CLAVE

Procesamiento de imágenes, Reconocimiento ópticos de caracteres, segmentación.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

ABSTRACT

The water, light and gas consumption meters are instruments to measure the amount of customer consumption in the service provided by the service companies.

Currently some companies continue to manage the information of the service provided in sheets manually or making use of applications where data is entered, perform this task in this way, create errors and time cats to collect and verify the data entered.

To make this serious change of the performance of an optical character recognition (OCR) that can be installed in a smartphone and thus streamline the process of capturing and validating the data.

Nowadays, smartphones have advanced with great progress, each one brings out new equipment with new structures and the quality of the images with good resolutions.

It was used 1300 images of consumption of electric energy meters as databases and about 3000 images of characters extracted by the original images, a protocol of photographic shots from a mobile application and the use of a free library as OpenCV was also proposed.

For the training, the training of the 3,000 images of the characters extracted and converted into a file was done and then in the classification reuse it.

And then run with a meter and the recognition of the characters with the use of the KNN classifier with the most suitable for the classification and that the results in the probability is 40% 60% and based on the recognition based on the accuracy in 80 % and with a neural network has had an accuracy of 72.7%.

KEYWORDS

Image processing, Optical character recognition, Segmentation.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

INTRODUCCIÓN

La presente investigación tiene como tema **"Identificación automática de caracteres numéricos en imágenes digitales de consumo de energía eléctrica"**, que describe los pasos como se captura las imágenes del consumo de los medidores, con la utilización de librerías libres y de multilenguaje, para utilización de técnicas de segmentación y el entrenamiento presentado por los investigadores.

Las fuentes bibliográficas resultan de meses de investigación en especial de los libros obtenidos en manera virtual, además con trabajos de investigación en torno a mi investigación. Este trabajo presenta los siguientes capítulos:

En el capítulo I se presenta el planteamiento de la investigación, el problema; delimitación, justificación e importancia, limitaciones y objetivos de la investigación.

En el capítulo II se abordan los aspectos teóricos relacionados con el tema de investigación, como los antecedentes del estudio, estado del arte, base teórica científica y definición de la terminología.

En el capítulo III se realiza el marco metodológico; que abarca el tipo y diseño de la investigación, población y muestra, hipótesis, variables, operacionalización; métodos, técnicas e instrumentos de recolección de datos, procedimiento para la recolección de datos, análisis estadístico e interpretación de los datos, criterios éticos y de rigor científico.

En el capítulo IV se realiza el análisis e interpretación de los resultados, en base a las tomas fotográficas, posteriormente se hizo el análisis de la aplicación basándose en los algoritmos del pre- procesamiento y la extracción del componente principal de la imagen.

En el capítulo V expondremos la propuesta de la investigación en seleccionar los dos métodos de reconocimiento de dígitos en imágenes digitales, y finalmente en el capítulo VI apreciaremos las conclusiones y recomendaciones de este trabajo de investigación.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

CAPITULO I: PROBLEMÁTICA DE LA INVESTIGACION

1.1. Situación problemática

En las empresas que brinda el servicio de energía eléctrica tienen como función el proceso de facturación de consumo de energía detallando en uno de los subprocesos más importante y crítico es de tomas de lecturas, donde el lecturista tiene como función de capturar los datos de numeración del consumos de energía eléctrica, se ha evidenciado que el problema se suscita por el mal ingreso de numeración de consumo de energía eléctrica del cliente, en que los mismos supervisores tiene que esperar las lecturas tomadas por lecturista para luego validarlas para ejecutar el proceso de facturación.

Tabla 1: Cuadro estadísticos de Reclamos 2016 Piura-Perú - Fuente: Gescom SAC

	2016											Total
	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	
Recibidos	1,318	2,100	1,573	2,385	2,030	1,349	954	835	727	772	433	14,476
Fundados	195	262	258	353	293	312	336	279	181	155	32	2,656
Infundados	1,123	1,838	1,315	2,032	1,737	1,037	618	556	546	617	401	11,820

Según **Flores (2006)** describe que el reconocimiento de caracteres en parte la parte de los resultados tiende a tener elevados porcentajes de fallos y un gran porcentaje de los costos de los recursos.

Según **Salas (2007)** describe que el reconocimiento de caracteres ante el rendimiento es baja cuando se trata de analizar la imagen con presencia de manchas o factores que no reconozca las palabras o número. Por lo que cuando se trata de trabajar con imágenes o documentos deteriorados para la digitación presenta carencias y deficiencias para el rendimiento en reconocimiento de caracteres.

También **Aguilera y Constenla(2010)** comenta que el reconocimiento de caracteres es una investigación con un gran importancia para el campo de reconocimiento de patrones y que en la actualidad los problemas de reconocimiento ópticos de caracteres (OCR) se encuentra

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

en resuelto a pesar que no reconoce el 100% y que es una de las áreas que se encuentra vigente hoy en día.

Según **Garrido (2010)** describe que el reconocimiento de dígitos o el reconocimiento ópticos de caracteres es una de las áreas de investigación más importantes por lo que el principal problema es la efectividad en el reconocimiento, por los motivos de la resolución, iluminación y otros factores catalogados en la imagen. También explica que para el reconocimiento de caracteres tiene un principal problema en que cada algoritmo que se usa para el reconocimiento tiende a la eficacia que se ve limitada por motivos a la precisión en la parte del tamaño, grosor con lo que se ve limitados ente buscar mayor reconocimiento o menor error en el reconocimiento al tener un nivel elevado de umbral de confiabilidad y el óptimos para realizar para cada técnica que se realiza en el reconocimiento.

1.2. Formulación de la problemática

¿Qué método sería más eficiente en la precisión para la identificación automática de caracteres en imágenes de numeración n de consumo de los medidores de energía eléctrica?

1.3. Delimitación de la investigación

En esta investigación busca comparar los diferentes métodos para la identificación automática de caracteres en las imágenes digitales de consumo de energía eléctrica, en nos vamos a enfocar solamente en los algoritmos propuesto por los investigadores donde tiene los resultados de los métodos de presión y compararlos.

La imagen solamente se plasmará en el consumo de los medidores mecánicos de energía eléctrica.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

1.4. Justificación e importancia de la investigación

La investigación tendrá una real importancia en la sociedad científica adicionando un aporte en el procesamiento de imágenes y en las máquinas de aprendizaje ya que es una de las ramas de la Visión Artificial que está en el mismo esquema de la Ciencia de la Computación e Inteligencia Artificial. Teniendo en cuenta que actualmente hay muchas investigaciones respecto al análisis, pero tiene más concentrado a los reconocimientos de caracteres en base a letras o en números para el proceso de en las imágenes a tiempo real, etc.

La investigación acerca del método de desarrollo para el reconocimiento de caracteres para la numeración de consumo de energía eléctrica nos permitirá en determinar la precisión en analizar los caracteres y la impresión de los resultados en los celulares móviles en que algún celular presenta con una pequeña dificultad por el tipo de formato en las imágenes que presenta, y también por la precisión de su autofocus en presentación de las imágenes en los celulares inteligentes. Además, el uso de los celulares móviles inteligentes o aplicativos de escritorio genera nuevas etapas.

Los beneficiarios serían las empresas que utiliza uso de medidores mecánicos o digitales para tener un mejor control de las validaciones de ingreso de datos como también mejoramiento de su proceso para un control total del trabajo.

1.5. Limitaciones de la Investigación

- En los medidores de energía eléctrica no presenta una posición fija para las tomas fotográficas.
- En los medidores de energía eléctrica solamente se usará estos modelos de medidores de energía eléctrica como: “SIEMENS”.
- Se tratará de usar un entrenamiento básico ya que el entrenamiento consume un gran recurso computacional por ello se tendrá en cuenta el uso de una computadora con características optimas en el hardware

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

y el uso de un celular para las tomas fotográficas.

1.6. Objetivo de la investigación

1.6.1. Objetivo General

- Evaluar método de identificación automática de caracteres de numeración de consumo en los medidores de energía eléctrica.

1.6.2. Objetivo Especifico

- Determinar el método de reconocimiento de caracteres.
- Elaboración plan de capturas de imágenes de consumo de medidores de energía eléctrica.
- Identificar características principales de las imágenes tomadas.
- Ejecutar los métodos.
- Evaluar los resultados tras la ejecución de los métodos

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Capitulo II: MARCO TEÓRICO

2.1. Antecedentes del estudio.

"Reconocimiento de caracteres mediante imágenes en contadores de gas en entornos reales" Bejarano y Sánchez (2015). La problemática se surgió con la interacción de la empresa de gas Madrileña Red de Gas, que su principal problema es la relación a lectura de contadores de gas bajó a su control, que dicho problema su principal objetivo es la automatización del procesado de lectura de los contadores de gas basado al uso de las aplicaciones de información. Durante el desarrollo del proyecto se ejecutaron algunas técnicas para el pre procesamiento como: Umbralizacion, suavizado, extracción de bordes (sobel) y la erosión para la mejora en la imagen para detectar, ya que el proyecto se utilizó el uso de Tesseract y la implementación de OpenCV porque es compatible con todo el leguaje que existe, para el reconocimiento se optó el uso de aprendizaje automático y el sin aprendizaje automático para la lectura del contador y detectaron que el uso de aprendizaje automático les puede ayudar cuando las imágenes son alejadas, ya que el otro tiene que acercar la cámara para el reconocimiento para la mejora de aciertos. Otro método para mejorar las tasas de acierto es utilizar otras técnicas de aprendizaje automático como pueden ser TBL (Aprendizaje Basado en Equipos) o HOG (Histograma de Gradientes Orientado). Su resultado fue positivos al realizar los métodos de aprendizaje.

"Implementación de un sistema de información para el reconocimiento de caracteres basado en la red neuronal perceptrón" Carranza (2014). Tiene como debido a que su implementación tendrá como función en entrenar la red para obtener los pesos y las vías de la red neuronal, detallando las técnicas del entrenamiento y maquetando junto con la arquitectura del perceptrón. En sus resultados concluyeron que las redes neuronales son importantes

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

para cualquier problema de clasificación siempre cuando haya data para el entrenamiento y el Método de Nguyen – Widrow es un buen método para la mejora y la aceleración de la fase del entrenamiento.

"Desarrollo de un sistema prototipo de reconocimiento de dígitos usando momentos invariantes" Gómez y Gutierrez (2011). En esta investigación su objetivo fue desarrollar un sistema prototipo de reconocimiento óptico de dígitos numéricos bajo condiciones específicas, utilizaron imágenes con caracteres numéricos y alfanuméricos, se utilizó el reconocimiento de una distribución normal multivariable con el teorema de Bayes y un vector de características invariantes. Los resultados de esta investigación determinan que los algoritmos implementados como la escala de grises, binarización, componentes conectados (determinar que parte de la imagen se encuentra el dígito para el proceso de pre-procesamiento, extracción y el reconocimiento de caracteres tiene la capacidad de generalización y que tuvo un 100% de efectividad en el experimento de matriz de covarianza semi-compartida con momentos Flusser N-fold.

"Reconocimiento Óptico de Caracteres (OCR)" Sánchez y Sadonís (2009). Describe que el reconocimiento óptico de caracteres (OCR) no se reconoce exactamente los caracteres de una determinada clase sino se puede distinguir entre un conjunto de elementos conteniendo en parte a la forma o el símbolo. Por lo que describe que cuanto más numeroso es el conjunto de las formas o símbolos tienda a tener mayor fallo en la clasificación.

2.2. Estado del arte.

"Clasificación De Dígitos Manuscritos De Imágenes Digitales" Tito y Silva (2015). Describe la problemática en el reconocimiento de caracteres es el uso de la clasificación es el más frecuente en la práctica, de esta forma, construiremos modelos que permita predecir la categoría de las instancias en función de una serie de atributos de entrada. Se utilizó técnicas para la extracción de características como la clasificación Análisis de Componentes Principales (PCA) con las dimensiones de la

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

imagen de 28X28 pixeles y el uso de la base de datos de dígitos de MNIST. En los resultados detallaron que el uso de una entrada de 92 componentes de la dataset se obtuvo una tasa de error de 5,12%, con otros métodos se verificaron este resultado:

- El uso de Algoritmo de clasificación de la distancia euclidiana se obtuvo una tasa de error de 1.15%.
- El uso de Algoritmo del vecino más cercano se obtuvo una tasa de error de 1.15%.
- El uso de Algoritmo del Support Vector Machine, se obtuvo una tasa de error de 1,04%.

"Reconocimiento de Números Manuscritos" Garbi, Mercado, Lanzarini y Russo (2007). En este trabajo determinan que el reconocimiento de números en textos manuscritos presenta un gran inconveniente debido a la presencia de algunos textos cursivos o texto que están juntos en una sola palabra.

Para el reconocimiento de números enteros manuscritos los autores desarrollan una Herramienta ICR (Reconocimiento inteligente de caracteres), integrando un clasificador basado en redes neuronales feedforward, en la etapa de pre procesamiento se adaptaron y normalizaron las imágenes, realizaron después una serie de pasos como la corrección del dígito si está inclinado, redimensionar las imágenes con el método de interpolación de vecinos más próximos ya que su resultado no genera nuevos píxeles, basándose en el filtrado de las imágenes con las proporciones de 39x27 pixeles, adelgazamiento, ensanchamiento y la reducción del tamaño basado a las proporciones de 13x9 pixeles para entrenar la red con un total de 7620 imágenes como bases de datos. De esta forma, se ingresa un número entero manuscrito formado por varios dígitos y se obtiene como resultado el reconocimiento de cada uno de los elementos que lo componen. Concluyeron su trabajo explicando que el uso de clasificadores basados en redes neuronales permite obtener una respuesta en un tiempo computacional breve con buenos resultados.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

"Reconocimiento de Dígitos Manuscritos Usando La Transformada Wavelet Continua en 2 Dimensiones y Redes Neuronales" Romero y Ruedin (2006). En esta investigación los autores exponen que el reconocimiento de caracteres tiene diferentes métodos, y que uno de los más detallados es el uso de redes neuronales, siendo importante la elección de la red neuronal y los tipos de dato que se debe procesar y sus características. Ellos plantearon realizar el pre procesamiento de los dígitos utilizando la transformada de wavelet debido a su excelente localización espacial y su buena localización de frecuencia. Se han realizado test con imágenes de dígitos de tamaño de 16x16 píxeles implicando el uso de la transformada de wavelet y el uso de un umbral para tener un resultado de valores binarios utilizando en la transformada una función llamada Mexican Hat direccional, también utilizaron la red de perceptrón multicapa con 160 neuronas para una capa oculta. La red fue entrenada con el algoritmo de backpropagation estocástico con momento para la obtención de los pesos y minimizar los costos. Basado a sus resultados tan tenido un porcentaje de patrones correctamente clasificado entre el 99.17 – 90.20 basado al conjunto de entrenamiento y testeo. Donde la conclusión detalla que el uso de transformada wavelet continúa en dos dimensiones que nos ha permitido entrenar una red neuronal feedforward multicapa como clasificador de dígitos manuscritos.

"Digit Classification on Signboards for Telephone Number Recognition" Yamaguchi, Nakano, Maruyama, Miyao y Hananoi, (2003). Hay un problema para el reconocimiento de dígitos de números telefónicos en carteles debido a la inclinación de los dígitos basado al punto de vista.

Para realizar esta adaptación del ángulo de inclinación se recogió un total de 1332 imágenes con 11939 dígitos, planteando su mecanismo de toma de fotografía en horas de día cuando estaba claro y nublado con 640x480 píxeles de escala de 8 bits.

Se trató de eliminar el ruido para obtener áreas candidatas de los números que se van a procesar utilizando el filtro de Roberts para

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

extracción de los bordes de los números y con eso la reducción del ruido disminuye, después de obtener los bordes se utilizó el filtro de binarización para tener la imagen etiquetada. Después de la reducción de ruido se realiza la corrección de la inclinación de las imágenes utilizando la Transformada de Hough, determinando el ángulo de la inclinación de la imagen, para determinar las áreas candidatas Y obtener la tasa de extracción de dígitos y reconocimiento de dígitos.

En los experimentos realizados con 1332 imágenes que contienen 11939 dígitos, se obtuvo un promedio de 99.2% de precisión para la extracción de dígitos y un promedio de 98.8% de precisión para el reconocimiento de dígitos.

2.3. Base teórica científica.

2.3.1. Imágenes Digitales.

2.3.1.1. Definiciones:

- **Aparicio (2014)** describe que la imágenes digitales son capturas mediante de dispositivos apropiados para su almacenamiento en una computadora.
- **Campos y Mundaca (2016)** describe que la imágenes digitales son capturas mediante de dispositivos apropiados para su almacenamiento en una computadora. También explica que los formatos de la imagen, se distingue en diferentes peso o formatos que emplea la imagen, como: JPG, PNG, GIF, TIFF.
- **Vilcherrez (2015)** describe que es la extracción de una parte del mundo real, como una representación bidimensional a partir de una matriz numérica.
- **Vassallo (2015)** describe que imágenes digitales es una representación de una imagen en lenguaje binario, es decir, una matriz numérica bidimensional de ceros y unos.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

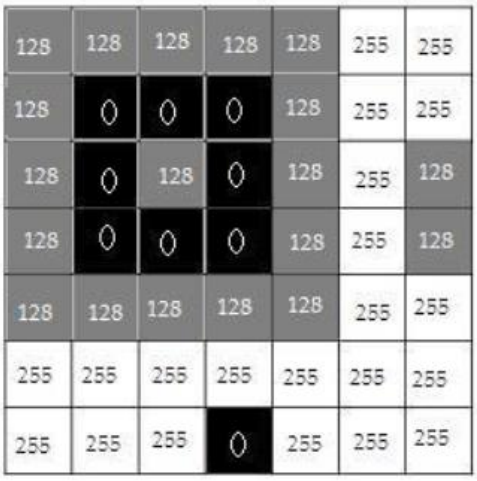

Figura 1:Representación de una imagen en una matriz. – Fuente: PROCESAMIENTO DE IMÁGENES DIGITALES UTILIZANDO DESCRIPTORES DE FORMA PARA LA IDENTIFICACIÓN DE DEFICIENCIAS NUTRICIONALES A NIVEL FOLIAR DEL CAFETO

2.3.1.2. Contenidos de imágenes digitales:

- **Resolución:** Es la representación en números de pixeles que representa una imagen.
- **Pixeles:** componente principal de color que es parte de una imagen.

2.3.2. Procesamiento de imágenes.

2.3.2.1. Técnicas o métodos:

a) Binarizacion

Según **Sebastián y Santiago (2011)** detalla que la binarizacion es la transformación una imagen en escala de grises a una imagen en dos colores: blanco o negro.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

$$I_2(x,y) = \begin{cases} 0 & \text{si } I_1(x,y) < T(x,y) \\ 255 & \text{en cualquier otro caso} \end{cases}$$

Figura 2: Esquema de binarizacion – Fuente: Opencv Documentación

b) Suavizado

Según **Caponetti y Castellano (2017)** detalla que el suavizado es un proceso para el desenfoque de una imagen para el procesamiento de imágenes. También se realizada para la eliminación del ruido.

Según **OpenCV (2017)** describe que el suavizado el principal objetivo es la reducción del ruido mediante de las técnicas del filtro: **Filtro Gaussiano, Filtro Bilateral, Filtro mediano, Filtro de caja normalizado.**

- **Filtro Gaussiano**

  Según **Caponetti y Castellano(2017)** describe que el filtro gaussiano es un operador de convolución bidimensional para desenfocar la imagen y recudir el ruido.

$$G(x,y,\sigma) = \left[\frac{1}{2\pi\sigma^2}\right] \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

Figura 3: Función del filtro gaussiano – Fuente OpenCV Documentación

OpenCV (2017)

- **Filtro bilateral**

  Según **Lélis (2015)** la ventaja del filtro bilateral es la conservación de la imagen tras la eliminación del ruido, en que cada componente de la imagen es muy importante, son más practicados en la parte de

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

ciencia médica y lo más complicado es el análisis de la distancia espacial con la diferencia de intensidad de pixeles al calcular.

Según **Chancafe y Mazabel (2016)** define como un promedio ponderado de pixeles adyacentes, de una manera muy similar a la convolución gaussiana.

$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|)\, G_{\sigma_r}(|I_{\mathbf{p}} - I_{\mathbf{q}}|)\, I_{\mathbf{q}},$$

Figura 4: Formula de Filtro Bilateral

Fuente: Detección automática de caries utilizando reconocimiento de patrones en placas radiográficas Chancafe y Mazabel (2016)

c) Canny:

Según **Vassallo (2015)** describe Canny es el más utilizado porque que permite la detección de bordes de una manera muy robusta y precisa.

Según **Caponetti y Castellano (2017)** describe que para la utilización de Canny sería adecuado pasar por el filtro gaussiano mediante la convulsión de la imagen.

Según **Lélis (2015)** describe que el informático John F. Canny realizo la técnica Canny para la optimización del detención de bordes con una baja tasa de error para la identificación correcta del borde en una imagen.

Según **OpenCV (2017)** detalla que el técnica Canny debes cumplir estos criterios:

- Baja tasa de error.
- Buena localización.
- Respuesta mínima.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

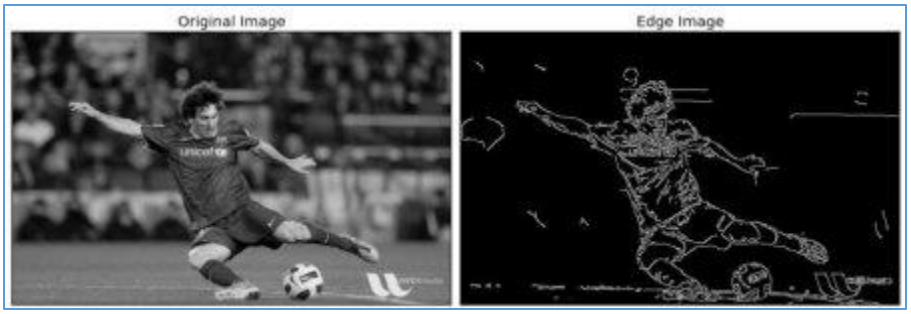

Figura 5: Modelo de Canny en una imagen.- Fuente: OpenCV Documentación
OpenCV (2017)

$$Edge\_Gradient\ (G) = \sqrt{G_x^2 + G_y^2}$$

$$Angle\ (\theta) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

Figura 6: Formula de Canny en OpenCV - Fuente: OpenCV Documentación
OpenCV (2017)

d) K vecinos más cercanos (K-NN)

Vassallo (2015) describe que es un algoritmo de aprendizaje supervisado y que no es un modelo fruto de aprendizaje con datos entrenado, sino que se ejecuta basado a los datos del test.

Vilcherrez (2015) es un clasificador se basa en la idea de que la predicción de la composición de un objeto debe ser el mismo que los objetos más similares.

e) Histograma Ecualización

Según **OpenCV (2017)** detalla que es un método de mejoramiento del constante de una imagen en cual tiene la función de estirar los rangos de intensidad de cada componente de la imagen.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Según **Lélis (2015)** detalla que la histograma ecualización hace un mejor uso de las intensidades que generan mejores contraste, utilizando una distribución acumulativa para reasignar el histograma a algo que se asemeja a una distribución uniforme.

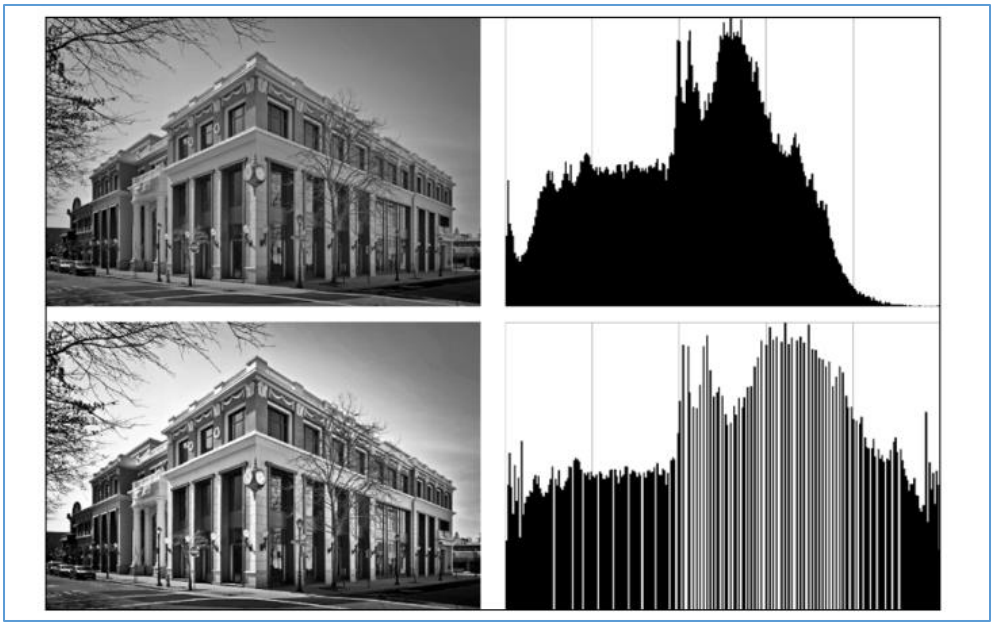

Figura 7: Ejecución de un imagen con histograma ecualizada - Fuente: OpenCV 3.0 Computer Vision with Java Lélis (2015)

2.3.2.2. Segmentación

- **Chancafe y Mazabel (2016)** describe que es la subdivisión una imagen en sus partes constituyentes u objetos. También suele ser el punto de partida de otro proceso de interpretación de más alto nivel.

- **Lucero y Saldaña (2016)** es el proceso de dividir o particional u n imagen en subregiones o regiones homogéneas. También se pude decir como número de regiones homogéneas que cada componente de división tiene una etiqueta única. Y que

los métodos más utilizados pata la segmentación son el método Otsu, Fuzzy C- Means y el K-Means.

- **Vassallo (2015)** describe que es un proceso que consiste en dividir o particional una imagen en sub regiones, componentes, partes u objetos, de manera que ayude en el análisis de la misma.

2.3.3. Redes Neuronales.

Casillas (2012) describe que objetivo principal de redes neuronales mediante de los algoritmos a realizar las tareas en las cuales el ser humano se desenvuelve muy bien. También explica que la red neuronal contiene 2 fases en que toda aplicación se distingue en la parte del aprendizaje o entrenamiento y la fase de la prueba, ya que la red neuronal verifique la eficacia real o error de los resultados.

- **Fase de Entrenamiento:**
  - Es la capacidad de aprender y actualizar basado a los cambios de los pesos simpáticos.
- **Fase de Prueba**:
  - Es el comportamiento final de una red neuronal que generaliza la salida de los valores esperados.

Chancafe y Mazabel (2016) describe que las redes neuronales son un método de resolver, para solucionar aquellas tareas de clasificación, identificación, diagnostico, optimización y la predicción de los datos.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

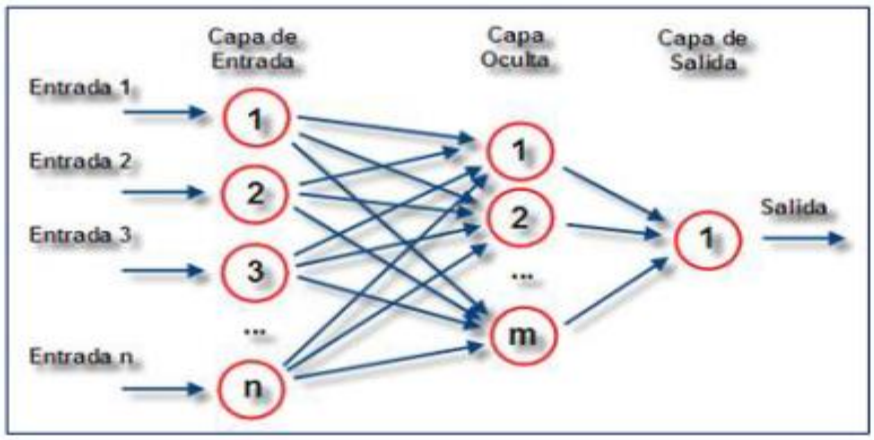

Figura 8: Estructura de una red neuronal multicapa.

2.3.4. KNN:

- **Vilcherrez (2015)** detalla que es un clasificador se basa en la idea de que la predicción de la composición de un objeto debe ser el mismo que los objetos más similares.
- **Lucero y Saldaña (2016)** detalla que es método de clasificación muy simple pero a la vez muy potente.
- **Vilcherrez (2015)** método que es un método de clasificación no paramétrico, que estima el valor de la función de densidad de probabilidad o directamente la probabilidad a posteriori de que un elemento.

COMIENZO
Entrada: $D = \{(\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_N, c_N)\}$
$\mathbf{x} = (x_1, \ldots, x_n)$ nuevo caso a clasificar
PARA todo objeto ya clasificado (x_i, c_i)
calcular $d_i = d(\mathbf{x}_i, \mathbf{x})$
Ordenar $d_i (i = 1, \ldots, N)$ en orden ascendente
Quedarnos con los K casos $D_{\mathbf{x}}^K$ ya clasificados más cercanos a $\mathbf{x}$
Asignar a $\mathbf{x}$ la clase más frecuente en $D_{\mathbf{x}}^K$
FIN

Figura 9: Pseudocódigo de KNN

Fuente: Clasificadores K-NN Abdelmalik, Inza y Larrañaga (2015)

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

2.4. Definición de la terminología.

a) **Computacional**. De la tecnología informática o relacionado con ella, o que utiliza los métodos o recursos de la informática.

b) **Distorsión**. Deformación de un sonido, una imagen, una señal, etc., producida durante su transmisión o reproducción.

c) **Imagen**. Arreglo bidimensional de píxeles con diferente intensidad luminosa (escala de gris).

d) **Inteligencia Artificial**: Disciplina de las Ciencias de la Computación que estudia, diseña métodos, algoritmos y sistemas que sean capaces de emular o imitar la inteligencia humana.

e) **Parámetro**. Es una variable que forma parte de los lenguajes de programación, métodos o algoritmos.

f) **Procesamiento de Imágenes.** Es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información.

g) **Transformación**. Conjunto de transiciones que define la sucesión de los estados.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Capitulo III: Marco metodológico

3.1. Tipo y diseño de la investigación

3.1.1. Tipo de la investigación:

Es Cuantitativa, aplicada y tecnológica, porque su naturaleza se infiere de conocimiento científico apoyado por las ciencias de la computación cuál resultado servirá para resolver problemas reales.

3.1.2. Diseño de la investigación:

El diseño de la investigación es de tipo Cuasi Experimental porque se verificará los cambios que surgen desde la variable teniendo en cuenta la realidad hoy en día en que las imágenes de pruebas son asignados al azar según las condiciones que lo proporciona dicha imagen.

3.2. Población y muestra

3.2.1. Población

Basado a una investigación (Islam, Islam, & Noor, 2017) plantea la estructura en global de los métodos para el reconocimiento ópticos de caracteres en un tabla, por lo que emplea de esta manera los métodos.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Phase	Description	Approaches
Acquisition	The process of acquiring image	Digitization, binarization, compression
Pre-processing	To enhance quality of image	Noise removal, Skew removal, thinning, morphological operations
Segmentation	To separate image into its constituent characters	Implicit Vs Explicit Segmentation
Feature Extraction	To extract features from image	Geometrical feature such as loops, corner points Statistical features such as moments
Classification	To categorize a character into its particular class	Neural Network, Bayesian, Nearest Neighborhood

*Figura 10: Paso o métodos para el reconocimiento ópticos de caracteres. – Fuente: **A Survey on Optical Character Recognition System** Islam, Islam y Noor (2017)*

3.2.2. Muestra

La utilización de los métodos pro conveniencia para la ejecución en la parte inicial y final del desarrollo como: binarizacion, dilatación, erosión, conversiones de imágenes, cambiar de tamaño segmentación, métodos de detención de contornos, clasificadores supervisado y no supervisados.

3.3. Hipótesis

El método más preciso es la red de neuronal perceptrón multicapa de para la identificación de caracteres de consumo de energía eléctrica.

3.4. Variables

Variable de estudio/investigación: Métodos de detección de caracteres.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

3.5. Operacionalización

Tabla 2: Referencia de los contenidos de operalización.

Término	Información Verdadera (TI)	Información Falsa (FI)	No Información Verdadera (TNI)	No Información Falsa (FNI)
Definición	marco informativo correctamente clasificado	Marco no informativo incorrectamente clasificado	marco no informativo correctamente clasificado	Marco no informativo incorrectamente clasificado

Tabla 3: Valores de la Operacionalización.

Variable	Dimensiones	Indicadores	Fórmula
Técnicas de detección de caracteres	Confiabilidad	Precisión (Prec)	$\frac{\sum TI}{\sum TI + \sum FI}$
		Sensibilidad (Sen)	$\frac{\sum TI}{\sum TI + \sum FNI}$
		F- Measure	$2 * \frac{Prec * Sen}{Prec + Sen}$

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

3.6. Métodos, técnicas e instrumentos de recolección de datos

3.6.1. Métodos de la investigación.

Para el método de la investigación se desarrollará bajo la técnica de instrumentos de recolección de datos desde la perspectiva metodológica cuantitativa.

3.6.2. Técnicas de recolección de datos.

Observación: Mediante esta técnica de observación se emplea en todo el proceso de desarrollo en la tesis en cual presentará en estos puntos:

Verificar los resultados de la implementación de los métodos planteados para la identificación automática de caracteres.

3.6.3. Instrumentos

a) Ficha de Observación

Para el desarrollo de tema de tesis se anotará los eventos presentado de la observación en que se hará una bitácora de pasos que se va desarrollar ya que la ficha de la observación es una relación entre la hipótesis y los hechos reales.

b) Bitácora

Para el desarrollo de tema de tesis se anotará los eventos detallando los avances y resultados preliminares que en su fin es desplegar qué acontecimiento surgió cuando se implementa los métodos planteados a la identificación en parte del pre procesamiento y el procesamiento de caracteres.

3.7. Procedimiento para la recolección de datos

Basado a la técnica de recolección de datos voy a detallar las especificaciones consideraciones:

Basado a la cantidad de 1300 imágenes de muestra se usará para realizar las respectivas características:

- Modelo de Medidor (Mecánico).
- Lugar de la Toma Fotográfica. (Ubicación de donde se tomó la foto).

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

- El modo de toma.
- Cantidades de caracteres de la enumeración de consumo del Medidor.

3.8. Análisis estadístico e interpretación de los datos

Se realiza un procesamiento estadístico descriptiva utilizando el software de Excel para la secuencia de la presentación, análisis e interpretación de los resultados finales de la investigación basado en los indicadores presentados en la operacionalización. En donde detalla los datos obtenidos de los resultados en promedio de la sumatoria, la varianza y los cuadros estadísticos, también se presentará en forma implícita los resultados para decretar el estimado con la veracidad.

3.9. Criterios éticos

- ✓ **Confidencialidad:** Los códigos de ética hacen énfasis ante la seguridad y la protección de los participantes como la identidad y como también los informantes de la investigación. También expresa como un anonimato en la identidad de los participantes detallando en un seudónimo entre los participantes.
- ✓ **Derechos de Autor:** Todo material usado para el proceso de tema de tesis estará referenciado y citados con sus respectivos autores como participante del trabajo.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

3.10. Criterios de rigor científico

Tabla 4: Criterios de rigor científico - Fuente: Elaboración Propia

Criterios	Características éticas del criterio
Originalidad	Se citará las fuentes bibliográficas de la información, con el fin de demostrar la inexistencia del plagio.
Validez	los datos obtenidos por el estudio o proyecto serán evaluados y analizados por los ingenieros especializados en el tema para decretar su veracidad,
Consistencia	La investigación representará material consistente y certificado por la comunidad científica
Fiabilidad	El proyecto cumplirá con las expectativas expresadas en su contenido y su implementación con el estándar y políticas para su desarrollo

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

CAPÍTULO IV: ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS

4.1. Resultados en tablas y gráficos

La interpretación de los datos se va especificara en según los resultados de la aplicación basado a los resultados de los algoritmos del pre-procesamiento y la extracción del componente principal de la imagen.

4.1.1. Tiempo del procesamiento.

Basado al tiempo de pre procesamiento y de la segmentación de las características para el entrenamiento hemos una comparación del resultado en el uso de bordes y los contornos para extraer las características por lo que ha constatado, usando el método de observación y el conteo de las imágenes que han segmentado los 5 caracteres.

Tabla 5: Comparación del algoritmo de detención de bordes con contornos.

Detectar de Bordes	Tiempo	N° imágenes	Imágenes Buen Segmentación	Positivo extracción %	Falso extracción %
Canny	10s	1300	10	0,8%	99,2%
Sobel	12s	1300	100	7,7%	92,3%
Otsu	20s	1300	800	61,5%	38,5%
Binary	16s	1300	500	38,5%	61,5%

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

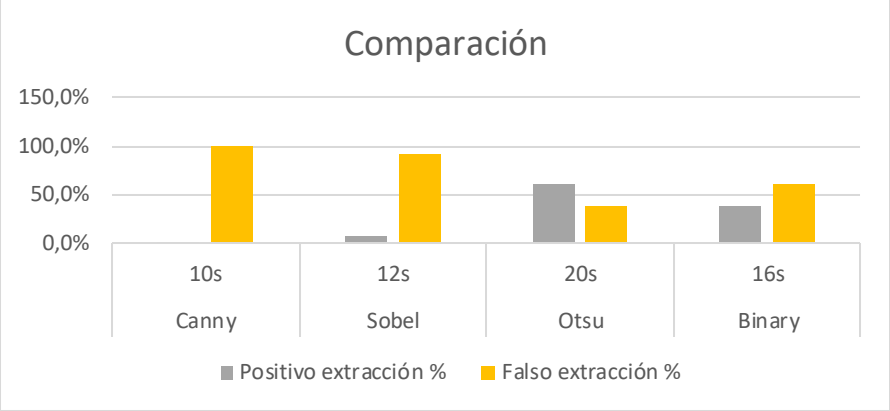

Figura 11: Estadística de la comparación de algoritmos.

4.1.2. **Tiempo del entrenamiento**.

Para el entrenamiento se hizo una estimación desde la primera numeración del entrenamiento hasta la última imagen del entrenamiento por lo que se hace una estimación.

Tabla 6: Tiempo del entrenamiento.

Imagen de caracteres	entrenar	Guardar
200	5 minuto	2 segundos
2000	50 minutos	3' 20 segundos
3000	1 hora 20 minutos	6' 20 segundos

4.1.3. **Tiempo del reconocimiento.**

Para el reconocimiento en parte KNN tenemos para el levantamiento de los datos del XML se demora entre 100 a 130 segundos y para el reconocimiento entre reconocer los dígitos se tuvo un resultado por imagen un promedio de 70 a 90 segundos. Y el entrenamiento de la red en el IDE Neuroph su tiempo de entrenamiento de 1 hora, pero a la ejecución de los resultados 1 segundo. A la utilización de un solo clasificador hemos tenido estos resultados:

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Tabla 7: Resultado de clasificación

Resultado	
KNN	
Información Verdadera (TI)	800
Información Falsa (FI)	200
No Información Verdadera (TNI)	200
No Información Falsa (FNI)	100
Precisión	80,00%
Sensibilidad	88,90%
F- MEASURE	84,20%
Redes neuronales	
Información Verdadera (TI)	800
Información Falsa (FI)	300
No Información Verdadera (TNI)	100
No Información Falsa (FNI)	100
Precisión	72,70%
Sensibilidad	88,90%
F- MEASURE	80,00%

4.2. Discusión de resultados

Para el protocolo de tomas fotográfica se tenido que hacer un trabajo en ambientes controlados por motivos de que algunos medidores tiene la deficiencia de la visualización.

Basado para el recorrido de tener los contornos de los medidores de energía eléctrica tenemos detalle de que el algoritmo de Otsu es un poco lento, pero procesa mejor que los otros identificadores, también que mucho se asimila el cercano el algoritmo de binarizacion por lo que es cuestión del manejo de los datos que se ingresa como también los datos como los procesa.

También el algoritmo para el clasificador se ha utilizado el KNN que ha

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

tenido los resultados por parte de uso de las observaciones en parte de la ejecución de las 1300 imágenes de medidores y con 3017 caracteres, en que al ejecutar el algoritmo de KNN no da una probabilidad de reconocimiento entre 40-60% de cada digito según la imagen y basado de los datos tenemos una precisión del 80% teniendo en cuenta que algunas imágenes se han reconocido por parte de 5 dígitos que tiene el consumo de medidor de energía y otros por parte a la cantidad de dígitos que ha reconocido, y en tanto en la red neuronal ha tenido una probabilidad de reconocimiento entre los números de consumo de 80 a 93% pero algunos se confundía con otros números ya que los datos de la red es por la data del entrenamiento, al analizar con algunas imágenes ha tenido una precisión del 72.7% .

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

CAPÍTULO V: PROPUESTA DE INVESTIGACIÓN

La propuesta que se emplea en este tema de investigación, es comenzar a seleccionar los dos métodos de reconocimiento de dígitos en imágenes digitales.

Posteriormente se empleará un protocolo de tomas fotográficas, considerando la distancia, la posición de colocación del medidor y el dispositivo de tomas fotográficas.

El primer paso de esta propuesta es a través del pre procesamiento de la imagen tomada considerando las siguientes características: reducción del tamaño, aumento del brillo, eliminación del ruido, suavizado de la imagen y la extracción de los contornos empleándose la técnica de detención de bordes para extraer cada dígito del contador del medidor de energía eléctrica, posteriormente pasarlo en un entrenamiento.

Teniendo en cuenta lo antes expuesto, se usará una red neuronal para la clasificación, posteriormente el testeo y, por último, pruebas de los resultados para determinar la detención automática.

5.1. Determinar el método de reconocimiento de caracteres.

Para la determinación de los métodos de reconocimiento de caracteres, se ha tenido a bien, analizar los diversos métodos que existen a fin de determinar un solo método que aplicará para el reconocimiento de caracteres.

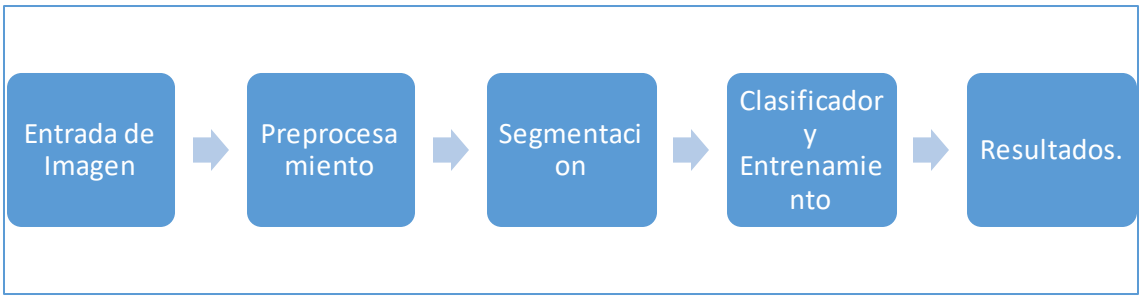

Figura 12: Método de reconocimiento de caracteres - Elaboración: Propia

5.1.1. Entrada de imagen

Es una técnica que se aplicará en la imagen de consumo en medidores de energía eléctrica a fin de obtener de ella la base de datos.

Es una fuente para el comienzo del pre procesamiento y la extracción de las características principales de la imagen, para que sea analizado y pasarlo a un entrenamiento para lograr un resultado.

Según **Wainschenker, Massa y Tristan (2011)** describe que la entrada de imágenes puede ser enfocada en el conocimiento del instrumento se utilizara las tomas y las especificaciones así como los resultados tomando en cuenta el tamaño, etc.

5.1.2. Pre procesamiento.

Es una técnica para extraer las características de una imagen para aplicar los algoritmos para el mejoramiento y la reducción de posibles bloqueos presentados en la entrada de la imagen.

Algoritmos que se emplea en la fase del pre procesamiento es:

- Reducción de la imagen.
- Escala de gris.
- Recusación de ruido.
- Suavizado de la imagen, etc.

5.1.3. Segmentación.

Esta fase permite conocer lo ya analizado en la fase de pre procesamiento, su principal funcionamiento es la región interés, con esta etapa se define que se quiere reconocer cuando se pasa hacia el entrenamiento.

5.1.4. Clasificador y Entrenamiento.

Es el paso fundamental para entrenar qué características tiene la imagen segmentada, ingresándolo nuevamente la imagen y su referencia.

5.1.5. Resultados

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Es el último paso, en la que se ingresara una imagen de un consumo de un medidor y ver el reconocimiento.

5.2. Elaboración plan de capturas de imágenes de consumo de medidores de energía eléctrica.

Para la realización de esta propuesta de investigación, es tener en cuenta el objetivo específico que se ha planteado y precisado en líneas arriba; es por ello que se ha creado una aplicación móvil para las capturas tomografías, implementado los módulos de OpenCV para tener un pequeño pre procesamiento, pero a nivel de escala de gris.

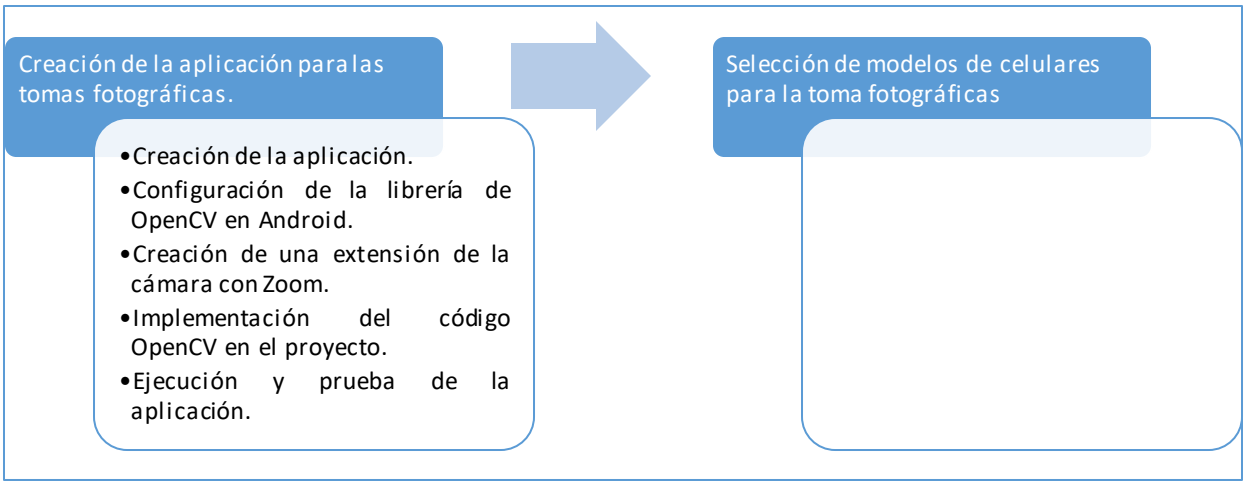

Figura 13:Fase de la elaboración de plan de capturas

5.2.1. Creación de la aplicación para las capturas fotográficas.

Para las capturas de las tomas fotográficas se realizó una aplicación móvil añadiendo los paquetes de OpenCV, que es una librería de Open Source, compatible con cualquier lenguaje de programación, dando a conocer que la importación del módulo de OpenCV en Android se debe realizar algunos pasos y verificar la comprobación de algunas características de los paquetes que contiene el OpenCv para Android.

En el anexo están los pasos de la importación de la librería, la codificación del proyecto y producto de estos resultados de la

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

programación del proyecto, tienen como resultado:

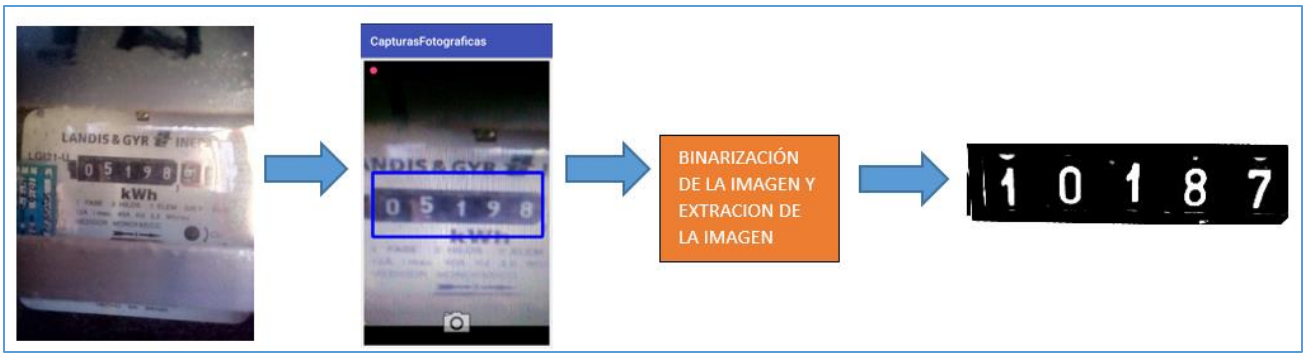

Figura 14: Método realizar el plan de capturas

En la figura se aprecia que con el aplicativo móvil tiene un contorno para la colocación del consumo del medidor de energía, a través del contorno se selecciona el icono de la cámara, teniendo en cuenta que en un proceso interno que se realiza la extracción de la imagen, posteriormente la conversión de la imagen a escala de grises.

El proceso de la escala de grises se da de esta forma.

$$\text{RGB[A] to Gray:} \quad Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

Figura 15: Formula para la escala de grises - Fuente: Documentación de OpenCV OpenCV (2017)

Con ese procedimiento se realiza la conversión de la imagen normal en escala de grises, después la extracción de la imagen completa a la imagen segmentada por el contorno, por lo que se utilizó una técnica de extraer el contorno y pasarlo a una nueva imagen.

5.2.2. Selección de modelos de celulares para la toma fotográficas

Para la realización de capturas de los contadores de medidores de energía eléctrica, se ejecutó en 2 equipos de celular detallando las características de la cámara y sus especificaciones.

- Especificaciones del Huawei P8:

Tabla 8: Especificaciones de Huawei P8. – Fuente: (Xataka, 2016)

DIMENSIONES FÍSICAS	PANTALLA	RESOLUCIÓN	PROCESADOR	CÁMARAS
144.9 x 72.1 x 6.4mm, 144 gramos	IPS LCD 5,2 pulgadas	1080 x 1920 píxeles (424 ppp)	HiSilicon Kirin 930, 4 núcleos a 2GHz y 4 a 1,5GHz	Principal de 13 MP con estabilización // Vídeo HD // Secundaria 5 MP
RAM	**MEMORIA**	**VERSIÓN SOFTWARE**	**CONECTIVIDAD**	**BATERÍA**
3 GB	16 / 32 / 64 GB (ampliable)	Android 6.0	LTE Cat 3, NFC, Bluetooth 4.1, Wi-Fi ac, Ant+, GPS, IR	2680 mAh polímero de litio (no extraíble)

- Especificaciones de Lenovo A2010

Tabla 9: Especificaciones de Lenovo A2010 - Fuente:(Lenovo, 2016)

DIMENSIONES FÍSICAS	PANTALLA	RESOLUCIÓN	PROCESADOR	CÁMARAS
66,6 x 9,98 x 130,5mm, 137 gramos	4,5 pulgadas	854 X 480 píxeles	MT6735m 64-bit Quad-Core 1.0GHz	5MP, foco fijo y flash LED, Foco fijo, 2M
RAM	**MEMORIA**	**VERSIÓN SOFTWARE**	**CONECTIVIDAD**	**BATERÍA**
1GB	16 / 32 GB (ampliable)	Android™ 5,1, Lollipop	LTE Cat 3, NFC, Bluetooth 4.0, Wi-Fi , GPS, IR	2050mAh polímero de litio.

Basado a las especificaciones de los celulares indicados en líneas arriba, se determinó que la aplicación tendrá estas siguientes características:

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

- Aumento del zoom para captar el consumo hasta el contorno.
- No utilización de flash para las tomas.
- Realizar 2 tomas por cada medidor de energía eléctrica.

5.3. Identificar características principales de las imágenes tomadas.

Basado a la toma realizada por el protocolo de capturas, las imágenes recolectadas han presentado estas características:

- La imagen con contenido de escala de gris para apaciguar el pre procesamiento de las imágenes.
- Formato se trabajará en modo *.JPG.
- Los tamaños de las imágenes son manejados por el tipo de celular como:
  - Huawei P8: 360 x 100 pixeles
  - Lenovo A2010: 250 x 60 pixeles

5.4. Ejecutar los métodos.

Para la realización de la ejecución del método, se ejecutará el proceso que se mencionó en el punto 5.1 de la propuesta de la investigación.

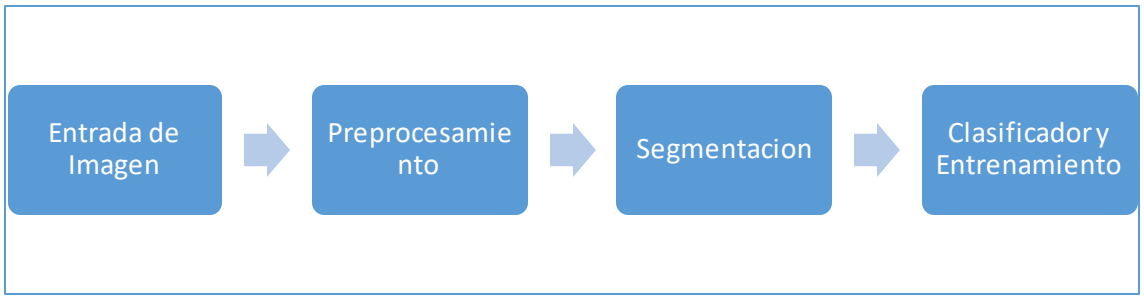

Figura 16: Método propuesto en el índice 5.1

Para la realización de los pasos de este método se ha tenido que crear un proyecto en Java con las librerías de OpenCv, que se detallará en los anexos de esta investigación (la instalación y la configuración de la librería de OpenCV al proyecto java escritorio).

Basado a ese proyecto creado se ejecutará los siguientes pasos del

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

método mencionado.

5.4.1. Entrada de la imagen:

Al ejecutar este paso del método se ha tenido que recolectar todas las fotos tomadas por los dos celulares que se mencionó anteriormente en el ítem "**5.2.2. Selección de modelos de celulares para la toma fotográficas"** en una carpeta, a partir de allí se hará la creación de la base de datos de todas las imágenes tomadas, se ha creído conveniente crear un nombre referencial a las imágenes para iniciar el proceso de la entrada de la imagen, teniendo en cuenta que la aplicación escritorio se utilizará la clase Mat del OpenCV para capturar la imagen, entonces así se realizará los siguientes pasos del método antes mencionado.

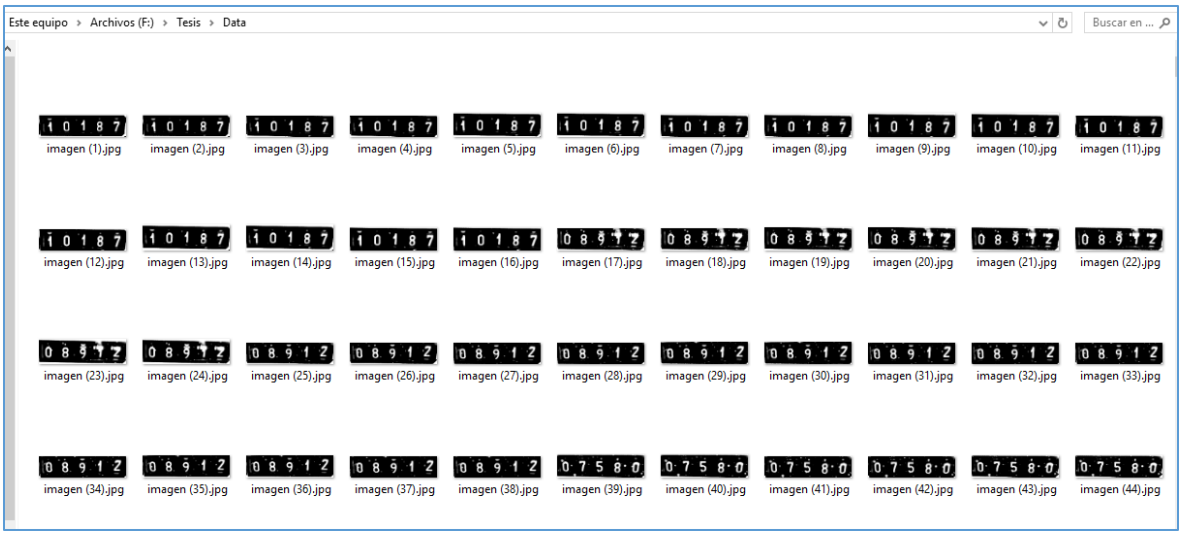

Figura 17: Carpeta de la bases de datos de la imágenes.

```
static String ubicacionorigen = "F:/Tesis/";
static String dataimagen = ubicacionorigen + "Data/";
Mat image = Imgcodecs.imread(dataimagen + "imagen (" + i + ").jpg",
        Imgcodecs.CV_LOAD_IMAGE_COLOR);
```

Figura 18: Codificación para la captura de la imagen

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

En el componente de "**imread**" se necesita dos parámetros; el primero es la ubicación de la imagen y el segundo es el formato de la imagen que deseas cargar, en el segundo parámetro es el CV_LOAD_IMAGE_COLOR que es el formato al estilo formato BGR **OpenCV (2017)**

5.4.2. Pre procesamiento:

Para ejecutar este procedimiento, se debe tener en cuenta las imágenes que han sido tomadas, luego ejecutar los siguientes algoritmos para el tratamiento de las imágenes, para iniciar la extracción de las imágenes que queremos analizar.

En la aplicación escritorio se realizará la ejecución del cambio del tamaño de la imagen, teniendo en cuenta que las imágenes deben estar en un solo estándar único del tamaño, detallado en el índice "**5.3. Identificar características principales de las imágenes tomadas",** donde se ha tenido dos diferentes tamaños al realizar las tomas fotográficas, por lo que se realizará un proceso de reducción o el aumento de un tamaño fijo para la ejecución de los demás algoritmos en el paso del pre procesamiento.

El algoritmo de reducción de imágenes se ejecutará de esta manera:

```
Imgproc.resize(image, image, new Size(250, 80));
```

Figura 19: Algoritmo para la reducción de la imagen en el proyecto .

Basado a esas características de las imágenes, de la imagen reducida se procede aplicar el método de conversión de escala de grises en las tomas fotográficas. Teniendo en cuenta que, al momento de convertir nuevamente a escala de gris, se procura más en la conversión de los pixeles cercanos a negros a negro

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

y pixeles cercanos a blanco a blanco. Basado a eso se hará el proceso de la dilatación y erosión de las imágenes para la extracción de líneas horizontales y verticales de las imágenes, para obtener un mejor control de cada componente y características de las imágenes.

```
Mat imageBlurr = new Mat(image.size(), CvType.CV_8UC4);
Mat imageBlurr1 = imageBlurr.clone();
Mat ones = Mat.ones(new Size(1, 1), CvType.CV_8U);
Imgproc.dilate(imageHSV, imageBlurr, ones);
Imgproc.erode(imageBlurr, imageBlurr, ones);
```

Figura 20: Algoritmo de ejecución de la dilatación y erosión de las imágenes.

Basado al proceso de la dilatación se usó el "CV_8UC4" que conforma el tipo de canales de 0 a 255 bits. Teniendo en cuenta a los resultados presentados por la dilatación y erosión de las imágenes.

Al tener la imagen procesada por los algoritmos de extracción de líneas, se realiza el proceso de filtro bilateral a una imagen para el proceso de reducción del ruido y el mantenimiento de los bordes del contenido de la misma.

Figura 21: A la izquierda se aprecia la imagen original, mientras que a la derecha se muestra la imagen con el filtro bilateral. Se puede apreciar el efecto de desenfoque

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

5.4.3. Segmentación

Para la realización de este proceso debemos tener en cuenta de los resultados de la imagen filtrada, se ejecutará el proceso de la detención de los contornos o bordes para ejecutar los siguientes algoritmos de Canny, Sobel, Otsu, Binarizacion; al momento de ejecutar estos algoritmos se ha observado un detalle, que el algoritmo del proceso Otsu junto con el algoritmo de localización de contornos reconoce los 5 dígitos que necesitamos para la extracción y almacenamiento antes de la ejecución del entrenamiento de los resultados extraídos.

En el proyecto se procedió hacer la ejecución de los 4 algoritmos mencionado para saber la diferencia y características de cada uno de los algoritmos.

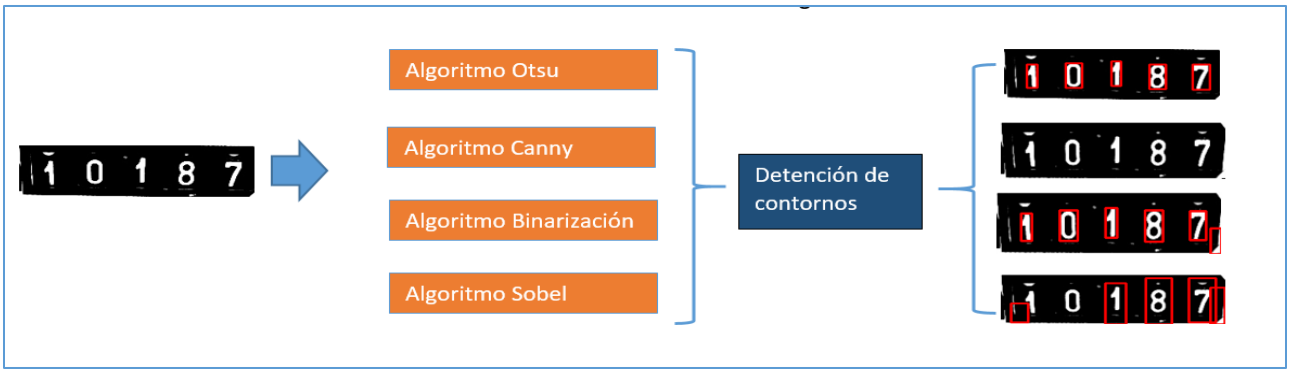

Figura 22: Técnicas de los algoritmos mencionado.

Basado a este proceso durante de la ejecución hemos tenido estos resultados.

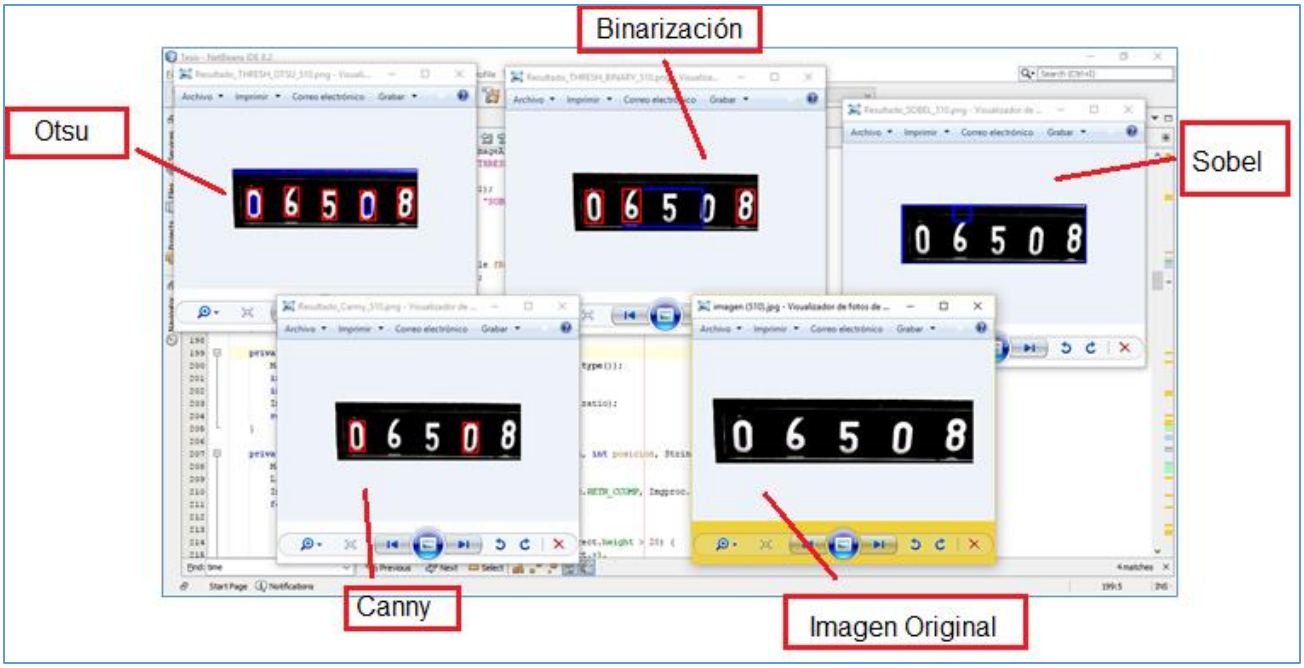

Figura 23: Resultados de la ejecución de los algoritmos.

Basado a los resultados presentado por el algoritmo Otsu, se realizará el proceso de extracción de los contornos segmentados de cada una de las imágenes de la base de datos, dándole un tamaño fijo de 30X54 y recortarlo para así almacenarlo en una carpeta diferente los contornos segmentados, después entrar en la parte de clasificación teniendo un solo estándar en el nombre de la imagen.

Aquí detallamos como se realizó el proceso.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

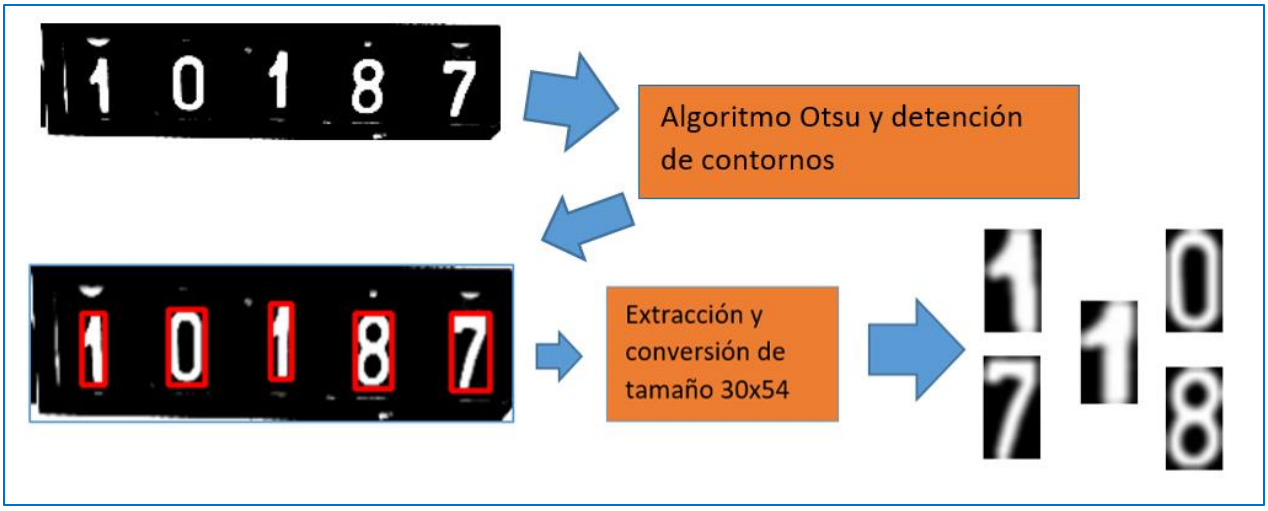

Figura 24: Proceso de segmentación y la extracción de dígitos.

Figura 25: Carpeta de las imágenes segmentados.

5.4.4. Clasificador y Entrenamiento

Para el proceso de clasificación y entrenamiento, se realizó un análisis de las imágenes que se ha segmentado para seleccionar las imágenes buenas y almacenarlo en otra carpeta, basado a eso se realizó un procesamiento de las imágenes segmentadas para hacer una conversión a data para el entrenamiento.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

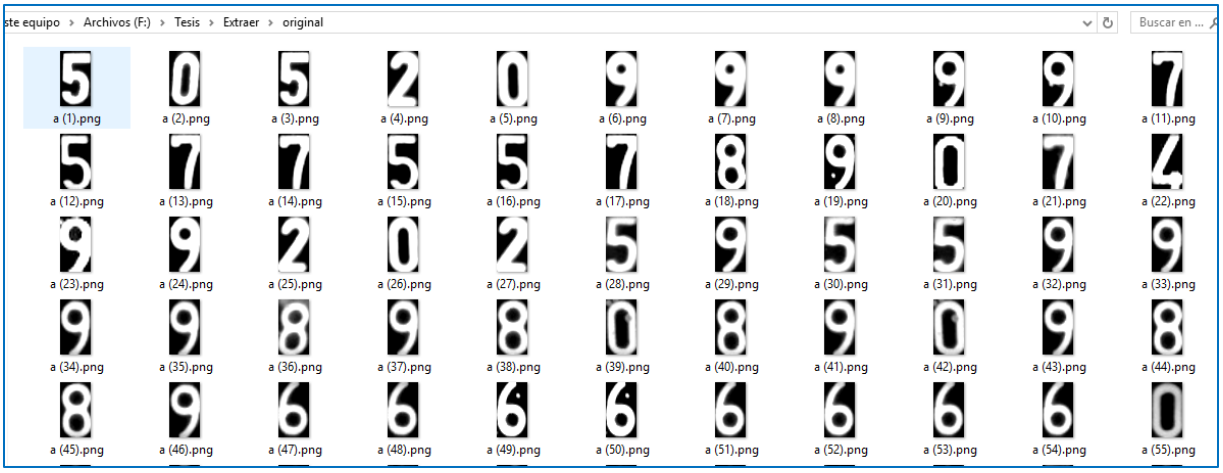

Figura 26: Carpeta de las imágenes buenas segmentadas.

El proceso se realizó de esta manera.

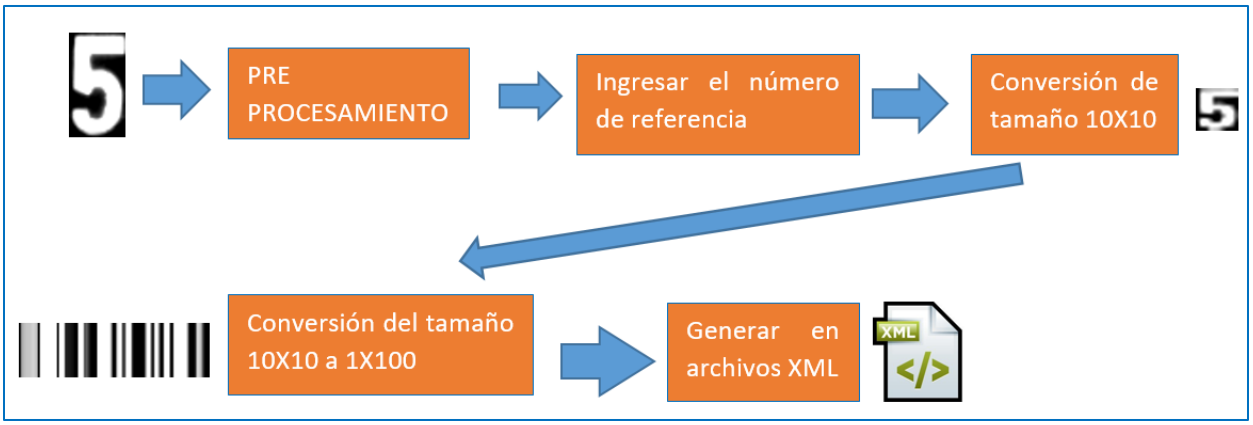

Figura 27: Proceso para el entrenamiento de la imagen.

En este proceso, la fase del pre procesamiento se realizó los procesos de eliminación de ruido, dilatación, erosión y la reducción de la imagen 10x10; para luego hacer el ingreso de número a que se refiere.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

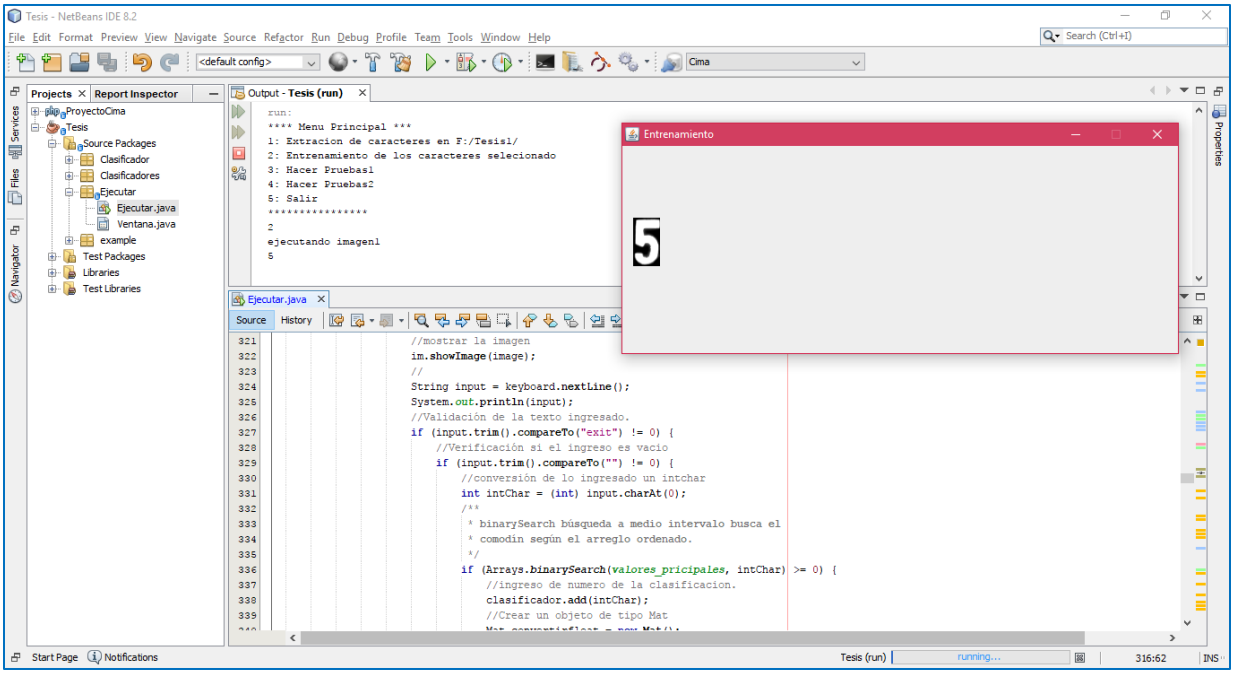

Figura 28: Proceso del entrenamiento.

En la imagen se hará el entrenamiento de cada dígito de la base de datos de las imágenes segmentadas, para realizar la conversión de la imagen 10x10 a imágenes 1x100 y así tener una data como sería la imagen con el numero referencial y luego almacenarla en 2 archivos XML, que hace referencia el primero la data de las imágenes y la segunda la data del numero referencial.

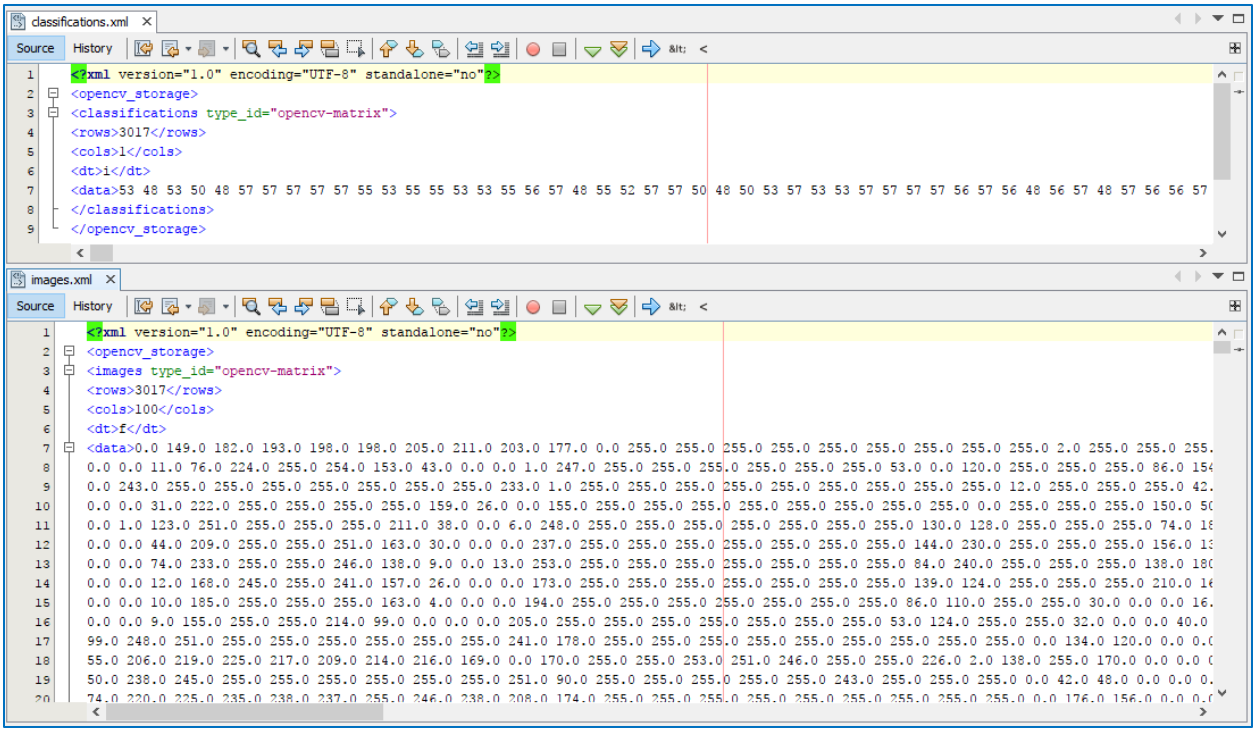

Figura 29: Datos del entrenamiento de cada digito y almacenarlo en un XML.

5.5. Evaluar los resultados tras la ejecución de los métodos

En los datos almacenados en el XML, se realizará la ejecución de una prueba para verificar los resultados de entrenamiento. Se debe cargar los archivos del XML al proyecto, e ingresar la imagen, hacer el pre procesamiento y la fase de la clasificación nos muestre el resultado del reconocimiento.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

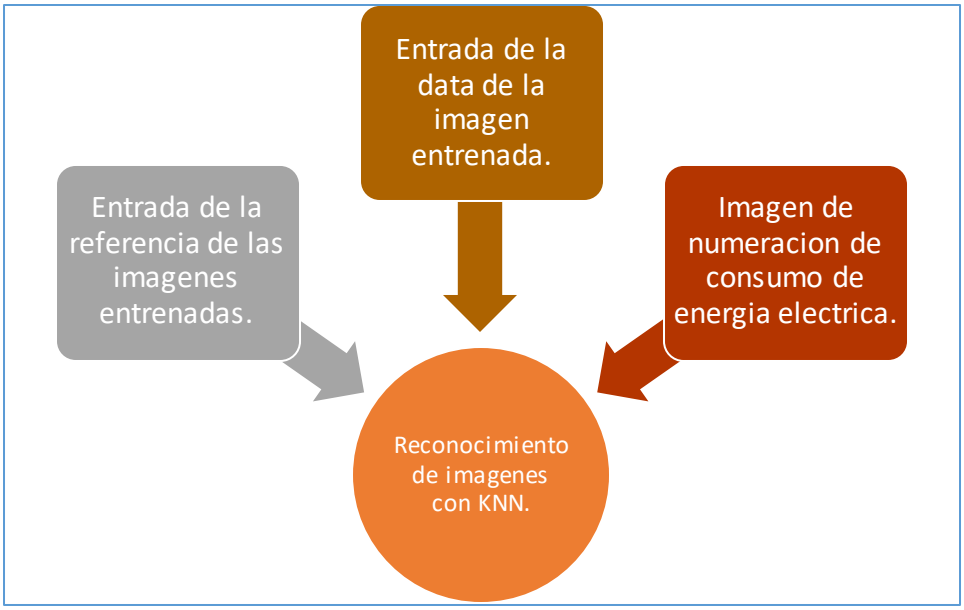

Figura 30: Modelo para evaluar los resultados.

Para realizar la ejecución del clasificar de KNN, vamos a tener en cuenta que los datos que se han analizado en las anteriores imágenes junto con los datos de referencia, será nuestra data para la utilización de los datos y así ingresamos una imagen para que los datos pueda ser reconocido y con eso dar la lógica del clasificador dando los resultados de la salida como también el porcentaje de probabilidad.

Lo primero es importar los datos del xml al aplicativo para tener una data referencial.

```
long inicio = System.currentTimeMillis();
label_data = ExtraerListLabel("src/Clasificador/classifications.xml");
double[][] ExtraerTrain = ExtraerTrain("src/Clasificador/images.xml");
Train = Mat.zeros(ExtraerTrain.length, 100, CvType.CV_32F);
for (int i = 0; i < ExtraerTrain.length; i++) {
    for (int j = 0; j < ExtraerTrain[i].length; j++) {
        Train.put(i, j, ExtraerTrain[i][j]);
    }
}
if (Train.dump().length() > 0) {
    JOptionPane.showMessageDialog(this, "Datos Cargados");
    //Ingresamos los datos de la data para luego ingresar en el método de KNN
    kNearest.train(Train, Ml.ROW_SAMPLE, Converters.vector_int_to_Mat(label_data));
    jTextField_CragaDatos.setText(((System.currentTimeMillis() - inicio) / 1000) + " segundos");
}
```

Figura 31: Método para subir los datos desde el XML.

Con eso el método **ExtraerTrain** vamos a coger los datos del xml para poder almacenarlo en una data con esos se ha hecho este algoritmo para poder tender los datos.

```
private synchronized static double[][] ExtraerTrain(String srcClasificadorclassificationsxml) {
    double[][] data_float = null;
    try {
        File fXmlFile = new File(srcClasificadorclassificationsxml);
        if (fXmlFile.exists()) {
            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
            Document doc = dBuilder.parse(fXmlFile);
            doc.getDocumentElement().normalize();
            //Ingresamos los datos del XML
            NodeList nList = doc.getElementsByTagName("images");
            for (int temp = 0; temp < nList.getLength(); temp++) { //recorer los datos
                Node nNode = nList.item(temp);
                if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                    Element eElement = (Element) nNode;//Contener los datos del xml a un Objecto
                    int row = Integer.parseInt(eElement.getElementsByTagName("rows").item(0).getTextContent());
                    int cols = Integer.parseInt(eElement.getElementsByTagName("cols").item(0).getTextContent());
                    String[] data = eElement.getElementsByTagName("data").item(0).getTextContent().split("\n");
                    data_float = new double[row][cols];//importar el número del color en el dato.
                    for (int i = 0; i < data.length; i++) {
                        String[] datal = data[i].split(" ");
                        for (int j = 0; j < datal.length; j++) {
                            data_float[i][j] = Double.parseDouble(datal[j]);//alamcenarlo en una sola data
                        }
                    }
                    Mat Train = Mat.zeros(row, cols, CvType.CV_32F);//Creacion de un objecto de tipo Mat.
                    for (int i = 0; i < data_float.length; i++) {
                        for (int j = 0; j < data_float[i].length; j++) {
                            Train.put(i, j, data_float[i][j]);//Añadir los datos a objecto Mat.
                        }
                    }
                }
            }
```

Figura 32: Método para extraer los datos del XML.

```
private List ExtraerListLabel(String srcClasificadorclassificationsxml) {
    List<Integer> data_label = null;
    try {
        File fXmlFile = new File(srcClasificadorclassificationsxml);
        if (fXmlFile.exists()) {
            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
            Document doc = dBuilder.parse(fXmlFile);
            doc.getDocumentElement().normalize();
            NodeList nList = doc.getElementsByTagName("classifications");//Reconocer el parametro de los datos
            for (int temp = 0; temp < nList.getLength(); temp++) {
                Node nNode = nList.item(temp);
                if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                    Element eElement = (Element) nNode;//Contener los datos en objecto de tipo Element.
                    int row = Integer.parseInt(eElement.getElementsByTagName("rows").item(0).getTextContent());
                    int cols = Integer.parseInt(eElement.getElementsByTagName("cols").item(0).getTextContent());
                    String[] data = eElement.getElementsByTagName("data").item(0).getTextContent().split(" ");
                    data_label = new ArrayList();
                    for (int i = 0; i < data.length; i++) {
                        data_label.add(Integer.parseInt(data[i]));//Alamacenar los datos del label en un lista.

                    }
                }
            }
        } else {
            data_label = null;
        }
    } catch (ParserConfigurationException | SAXException | IOException ex) {
        ex.printStackTrace();
    }
    return data_label;
```

Figura 33: Método para extraer los datos de referencia.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Basado a eso vamos a ingresar la imagen para poder hacer el pre procesamiento de la imagen y luego de eso tener la data de la imagen segmentada para poder ingresar al KNN para que nos de los resultados como también sacar la probabilidad de los datos.

```
Imgproc.rectangle(recorte, new Point(rect.x, rect.y), new Point(rect.x + rect.width, rect.y
        + rect.height), new Scalar(255, 0, 0), 2);
Mat result = image.submat(rect);
Imgproc.resize(result, result, new Size(10, 10));
if (result.channels() == 3) {
    Imgproc.cvtColor(result, result, Imgproc.COLOR_BGR2GRAY);
}
Mat convertirfloat = new Mat();
result.convertTo(convertirfloat, CvType.CV_32F);
Mat response_train = convertirfloat.reshape(1, 1);
//Ingresamos los datos de la imagen y luego vamos a ingresar el número de vecinos cercanos que es el 10
//kNearest.findNearest("Data ingresada", "número de vecinos", "Data de salida");
float p = kNearest.findNearest(response_train, 10, new Mat());
int binarySearch = Arrays.binarySearch(valores_pricipales, (int) p);
//kNearest.predict --> te arroja la probabilidad de los datos analizados.
resultado += "Segmentación:" + cantidad + "[- Probalidad:" + (kNearest.predict(response_train))
        + "]" + " - Numero: " + binarySearch + "\n";
arreglo += binarySearch + "-";
Imgproc.putText(recortel, "(" + cantidad + ")", new Point(rect.x, rect.y + rect.height)
        , Core.FONT_HERSHEY_SIMPLEX, 0.5, new Scalar(255, 0, 0));
```

Figura 34: Método para hacer el uso de KNN en el sistema.

Y con ese contenido vamos ya ejecutar el sistema para hacer el reconocimiento de los datos con el clasificador KNN

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

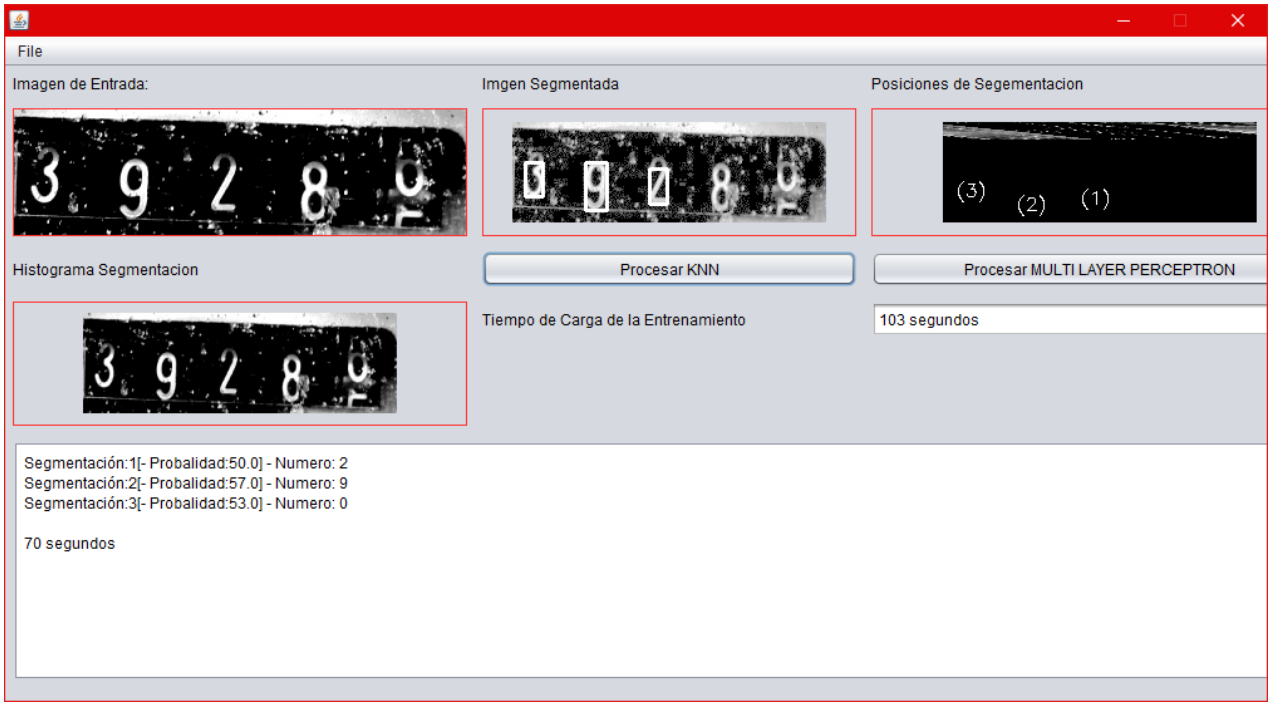

Figura 35: Reconocimiento de los números con un número error.

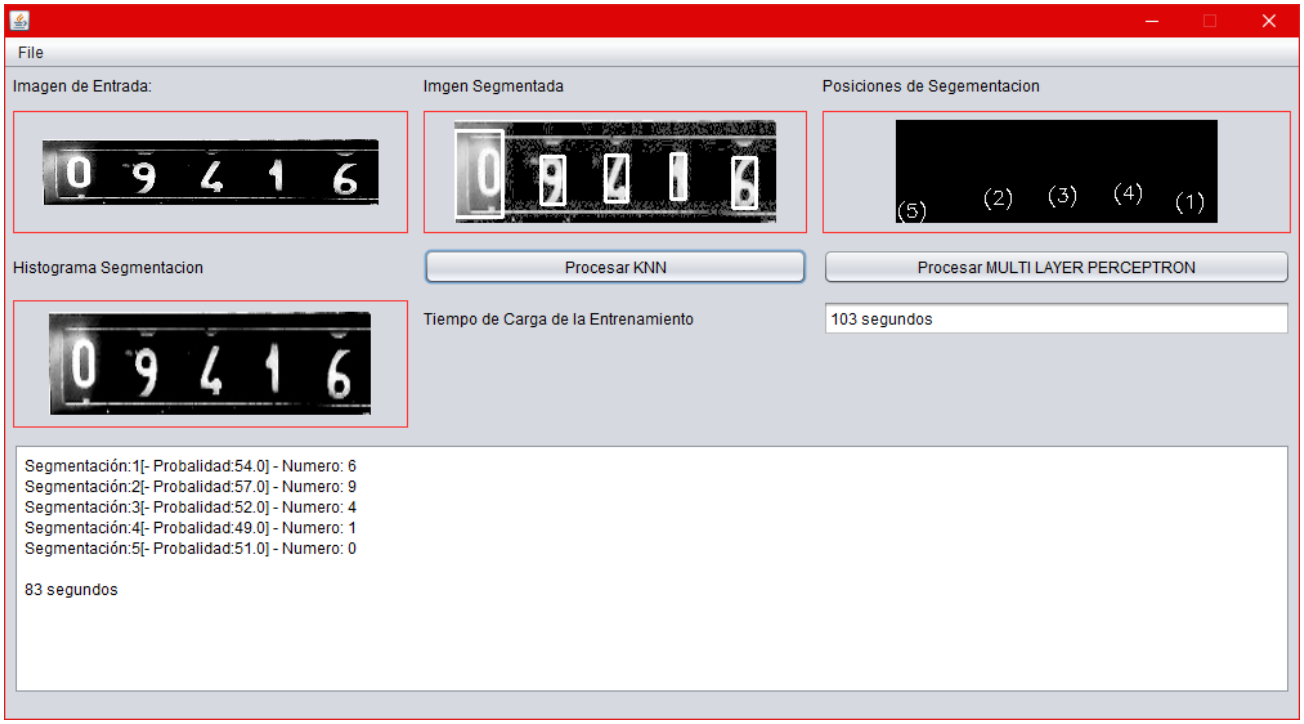

Figura 36: Reconocimiento total de los caracteres.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

De estos resultamos hemos tenido como resultado que en parte del reconocimiento basado a la probabilidad del reconocimiento tenemos entre 40% al 60 % según el algoritmo de KNN y por parte de recolección del reconocimiento en manera visual y estadísticos de los resultados hemos tenido entre 80% de las 1300 imágenes de prueba.

5.6. Entrenamiento para la Multi Layer Perceptron con Neuroph

Para la realización de otro entrenamiento y ver los resultados hemos tenido que utilizar un software llamado Neuroph por lo que hemos tenido que hacer un entrenamiento para almacenar los datos de la imagen como también almacenar el dato referencial de la imagen como hemos visto para el entrenamiento del KNN, teniendo en cuenta que aquí ya no vamos a utilizar un XML sino un archivo plano para almacenar cada dato y luego procesar en el software.

```
private static void SacarArchivo2(List<Integer> clasificador, Mat imagen_train, String filename) {
    long TInicio, TFin;
    TInicio = System.currentTimeMillis();
    System.out.println("Convertir los datos en block de notas");
    String dataClassifications = "";
    //recorrer todo las imagen y almacenar en una variable.
    for (int i = 0; i < imagen_train.rows(); i++) {
        for (int j = 0; j < imagen_train.cols(); j++) {
            double[] temp = imagen_train.get(i, j);
            dataClassifications += (int) temp[0] + ",";
        }
        dataClassifications += clasificador.get(i) + "\n";
    }
    //Configuracion de almacenamiento en el archivo plano
    FileWriter fileWriter = null;
    StringTokenizer st = new StringTokenizer(dataClassifications, "\n");
    PrintWriter pw;
    try {
        fileWriter = new FileWriter(filename);//Ubcamos el sitio del archivo del xml para la modificación
        pw = new PrintWriter(fileWriter);
        while (st.hasMoreTokens()) {
            String line = st.nextToken();//Insertamos los datos de entrada
            pw.println(line);//imprimimos los datos al archivo plano
        }
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        try {
            if (null != fileWriter) {
                fileWriter.close();
            }
        } catch (IOException e2) {
```

Figura 37: Programación del almacenamiento de los datos para la red neuronal.

Ya teniendo en cuenta de esto tenemos como resultado esta formación de los archivos al entrenar con la misma cantidad de imágenes del KNN.

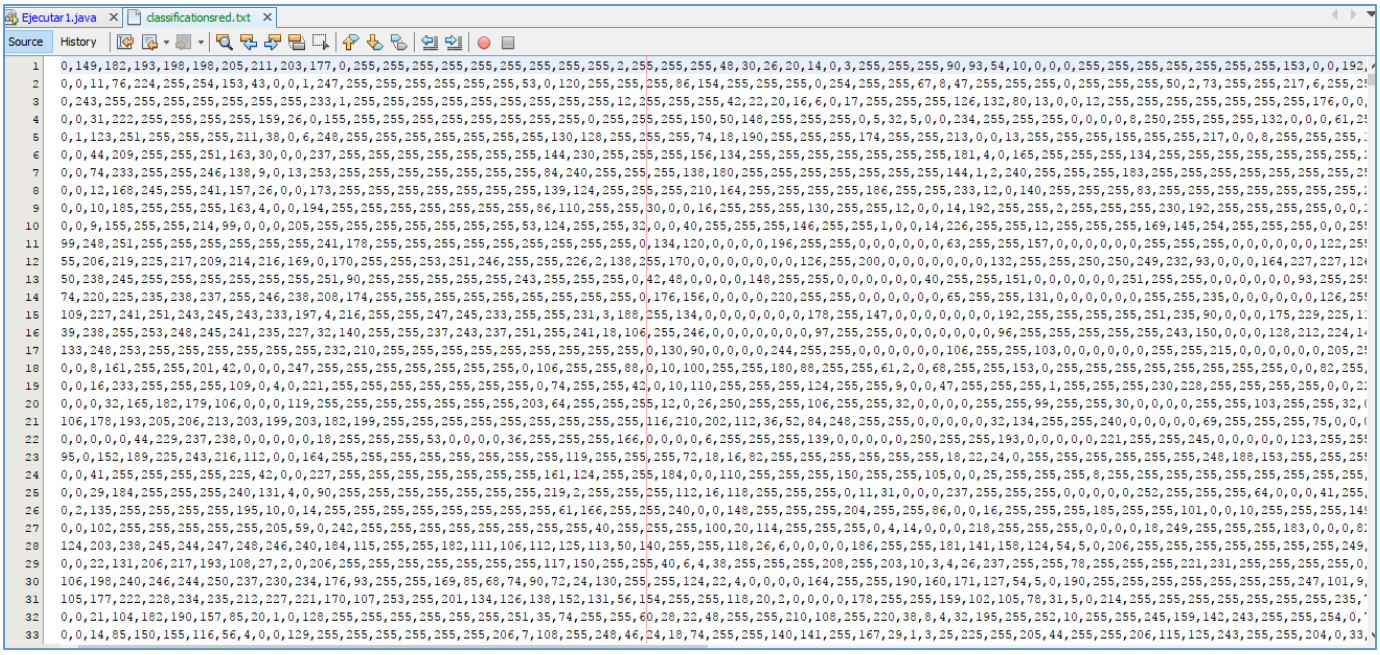

Figura 38: Resultados del entrenamiento y almacenar en un archivo plano.

Teniendo en cuenta vamos a pasar esos resultados a un Excel para la conversión de datos a referencial a un dato principal para que la red entienda que numero pertenece, porque en el software te pide la cantidad de datos de entrada y cantidad de datos de salida.

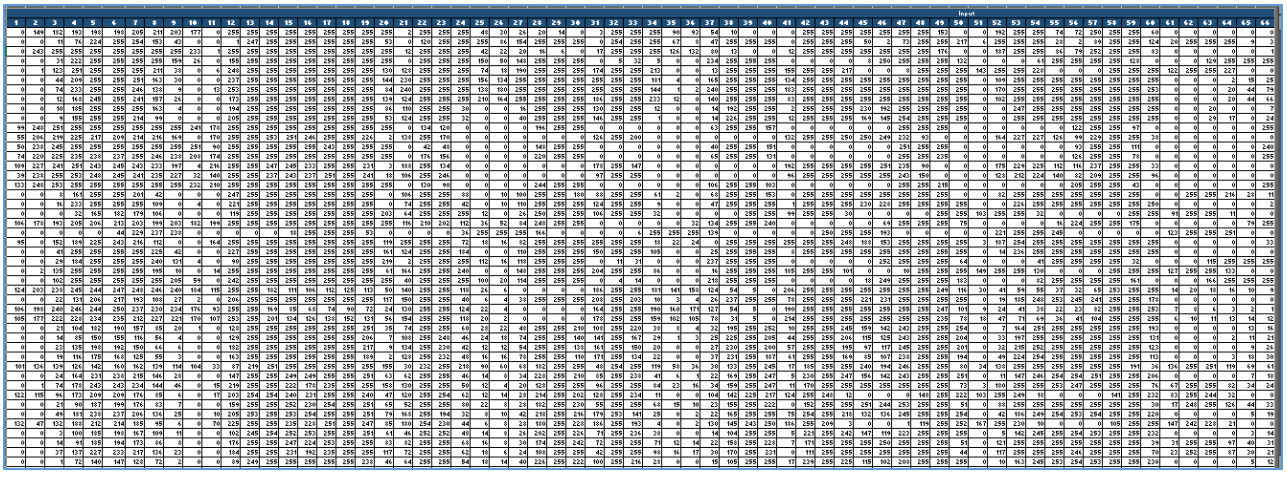

Figura 39: Datos de la imagen desde el archivo plano al Excel

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

																				ouput										Referencia	
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	0	1	2	3	4	5	6	7	8	9	Dato	Nº
44	0	255	255	255	255	255	255	255	155	0	0	0	36	203	209	177	71	6	0	0	0	0	0	0	1	0	0	0	0	53	5
38	0	249	255	255	255	255	255	255	107	1	0	0	101	233	251	216	102	4	0	0	1	0	0	0	0	0	0	0	0	0	48	0
58	8	255	255	255	255	255	255	255	192	4	0	0	88	229	237	214	146	16	0	0	0	0	0	0	0	1	0	0	0	0	53	5
0	228	255	255	255	255	229	217	201	148	21	255	255	255	255	255	255	255	255	255	116	0	0	1	0	0	0	0	0	0	0	50	2
255	26	255	255	255	163	64	148	255	255	255	0	30	219	255	255	255	255	255	254	51	1	0	0	0	0	0	0	0	0	0	48	0
98	0	0	0	2	184	255	255	255	117	0	0	0	0	158	255	255	213	30	0	0	0	0	0	0	0	0	0	0	0	1	57	9
66	0	0	0	3	172	255	255	255	97	0	0	0	0	183	255	255	217	38	0	0	0	0	0	0	0	0	0	0	0	1	57	9
148	0	0	0	0	79	255	255	255	187	0	0	0	0	100	255	255	255	118	0	0	0	0	0	0	0	0	0	0	0	1	57	9
120	0	0	0	0	0	106	255	255	130	0	0	0	0	22	250	255	255	44	0	0	0	0	0	0	0	0	0	0	0	1	57	9
106	0	0	0	0	0	148	255	255	129	0	0	0	0	46	251	255	255	40	0	0	0	0	0	0	0	0	0	0	0	1	57	9
0	0	0	0	0	34	255	255	93	0	0	0	0	0	0	42	255	255	21	0	0	0	0	0	0	0	0	0	1	0	0	55	7
255	0	126	175	93	0	0	0	200	255	198	0	24	237	248	240	240	246	251	149	0	0	0	0	0	0	1	0	0	0	0	53	5
0	0	0	0	0	0	255	255	123	0	0	0	0	0	0	0	255	255	44	0	0	0	0	0	0	0	0	1	0	0	0	55	7
0	0	0	0	0	44	255	255	68	0	0	0	0	0	0	52	255	255	2	0	0	0	0	0	0	0	0	1	0	0	0	55	7
255	0	164	191	68	0	0	0	243	255	183	0	74	253	255	255	252	255	255	140	0	0	0	0	0	0	1	0	0	0	0	53	5
255	0	139	211	137	0	0	0	183	255	242	0	12	238	255	255	255	255	255	174	0	0	0	0	0	0	1	0	0	0	0	53	5
0	0	0	0	0	113	255	255	5	0	0	0	0	0	0	116	255	251	0	0	0	0	0	0	0	0	0	0	1	0	0	55	7
255	0	225	255	255	15	0	9	255	255	255	0	0	71	255	255	255	255	255	206	0	0	0	0	0	0	0	0	0	1	0	56	8
0	0	0	0	0	0	235	255	255	4	0	0	0	0	28	255	255	255	0	0	0	0	0	0	0	0	0	0	0	0	1	57	9
255	81	255	255	75	0	0	0	6	255	255	7	255	255	255	234	207	226	255	255	255	1	0	0	0	0	0	0	0	0	0	48	0
0	0	0	0	13	225	255	255	103	0	0	0	0	0	0	242	255	255	14	0	0	0	0	0	0	0	0	1	0	0	0	55	7
244	180	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0	214	255	255	10	0	0	0	0	1	0	0	0	0	0	52	4
144	0	0	0	0	11	255	255	255	243	0	0	0	3	144	255	255	255	188	0	0	0	0	0	0	0	0	0	0	0	1	57	9
60	0	0	0	0	71	255	255	255	96	0	0	0	0	170	255	255	245	20	0	0	0	0	0	0	0	0	0	0	0	1	57	9
0	255	255	255	255	245	213	209	204	131	0	255	255	255	255	255	255	255	255	255	27	0	0	1	0	0	0	0	0	0	0	50	2
255	13	255	255	255	143	69	166	255	255	247	0	5	175	255	255	255	255	255	161	0	1	0	0	0	0	0	0	0	0	0	48	0
0	255	255	255	255	255	219	207	195	162	45	255	255	255	255	255	255	255	255	255	176	0	0	1	0	0	0	0	0	0	0	50	2
252	224	255	255	203	125	149	242	255	255	121	80	246	255	255	255	255	255	224	90	2	0	0	0	0	0	1	0	0	0	0	53	5

Figura 40: Modelo de la referencia en 0 y 1 para la red

Una vez que tenemos esos datos vamos a crear la red y los datos en el software Neuroph.

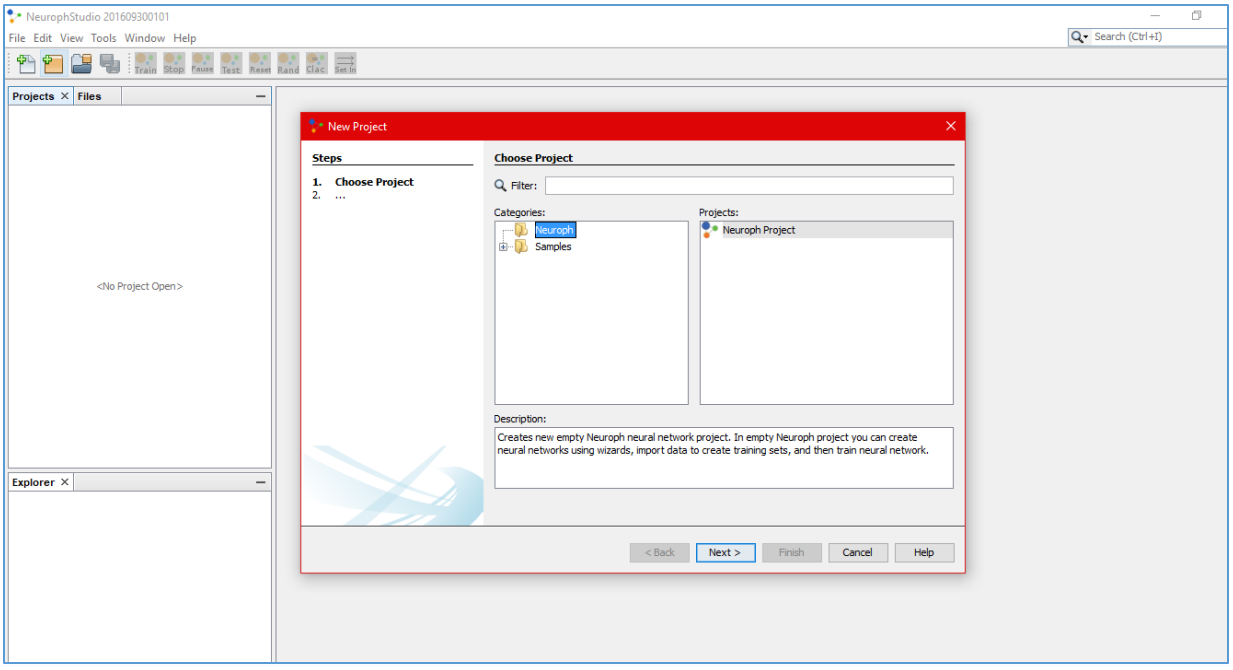

Figura 41: Creación de la aplicación en el IDE Neuroph.

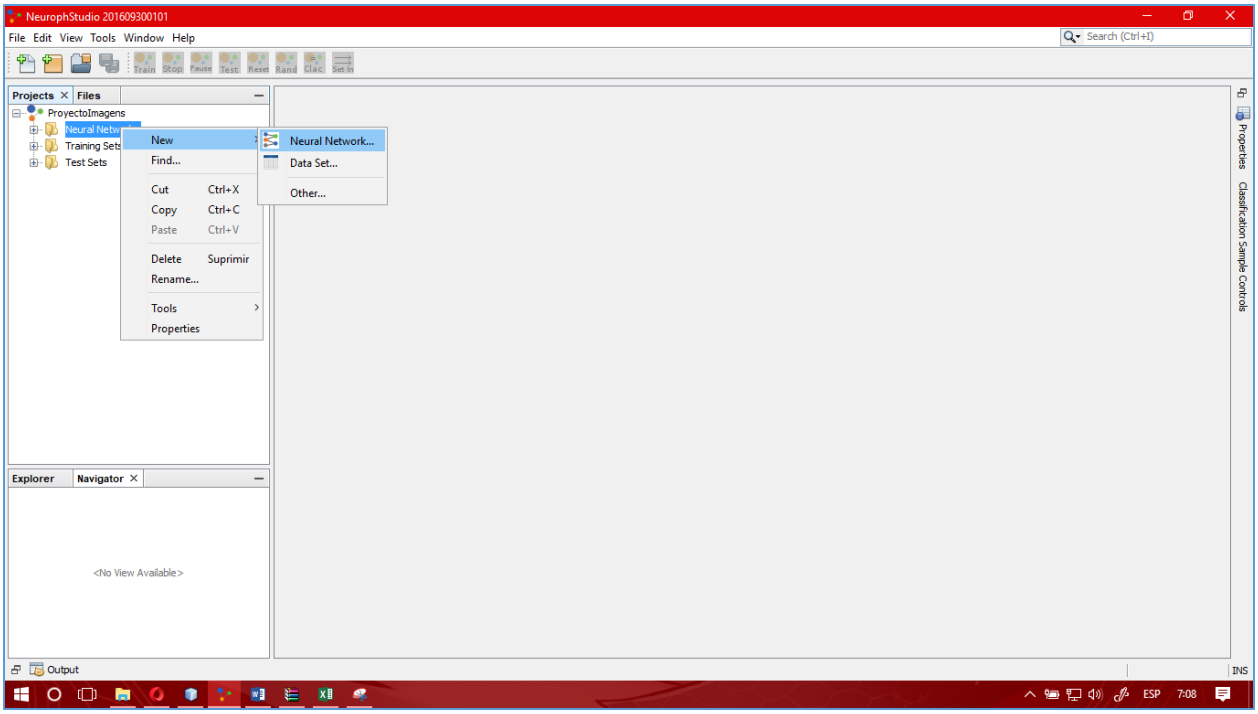

Figura 42:Creación de la red y los datos para la red.

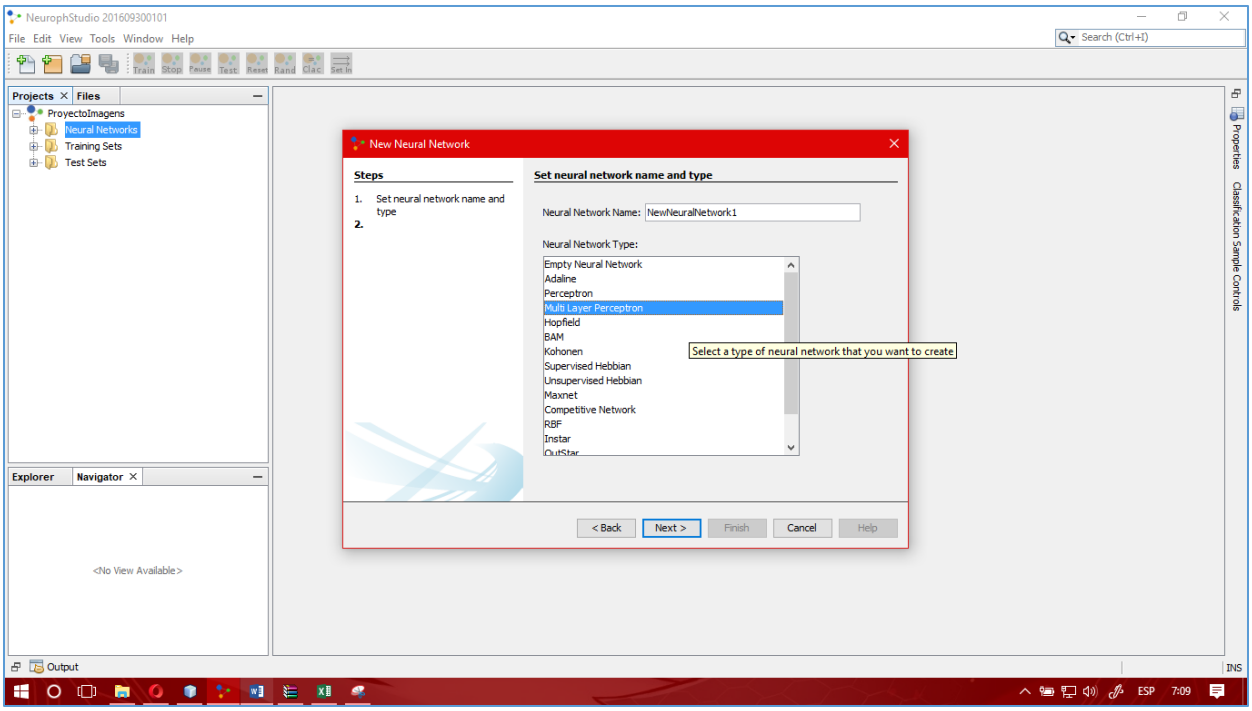

Figura 43: Seleccionar el tipo de red.t

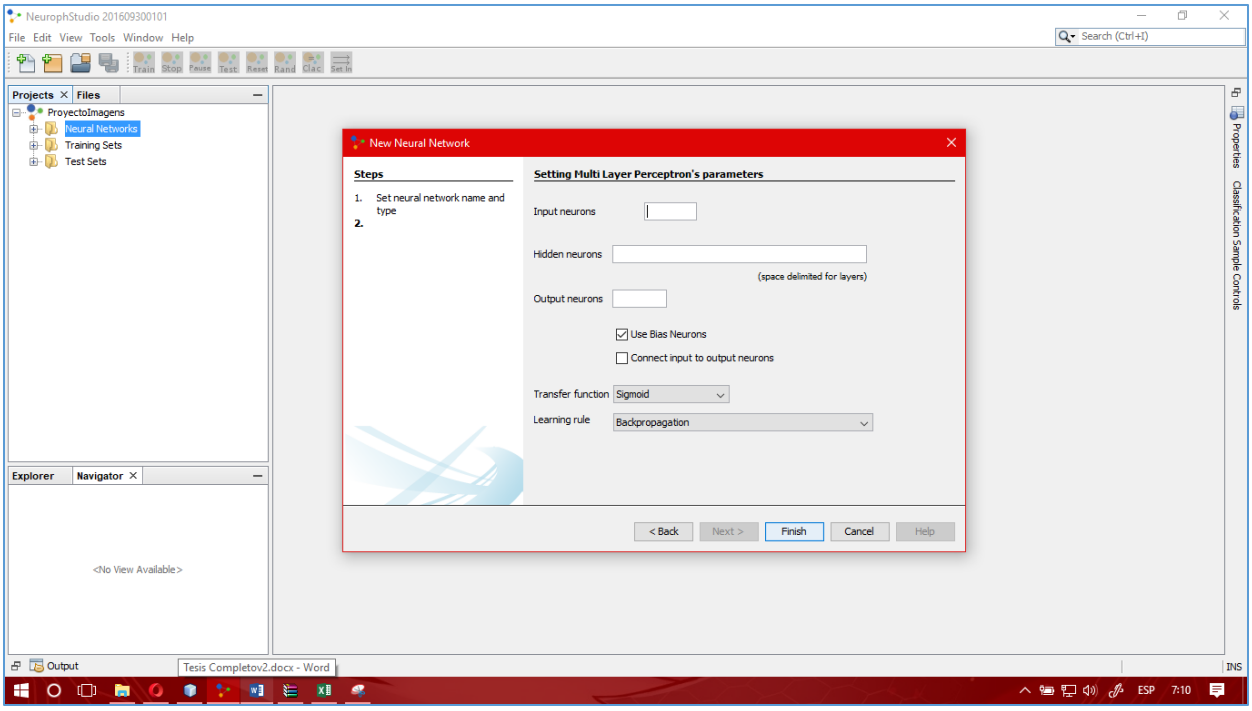

Figura 44: Ingresamos la cantidad de entrada y salida de la red.

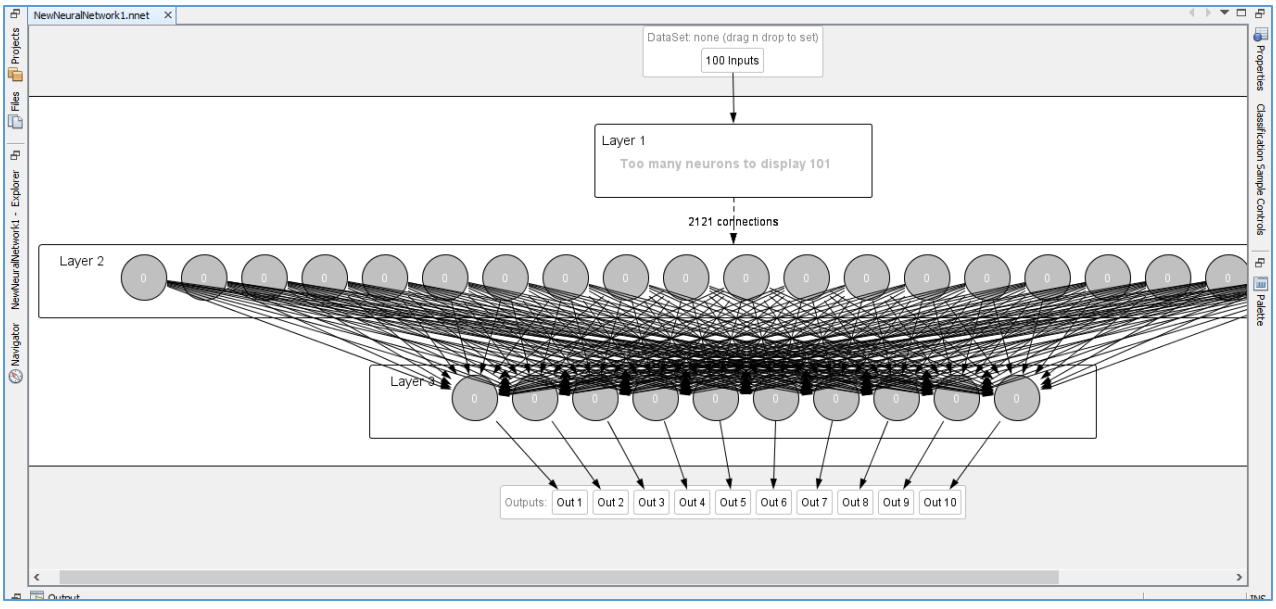

Figura 45: Formato de la creación de la red.

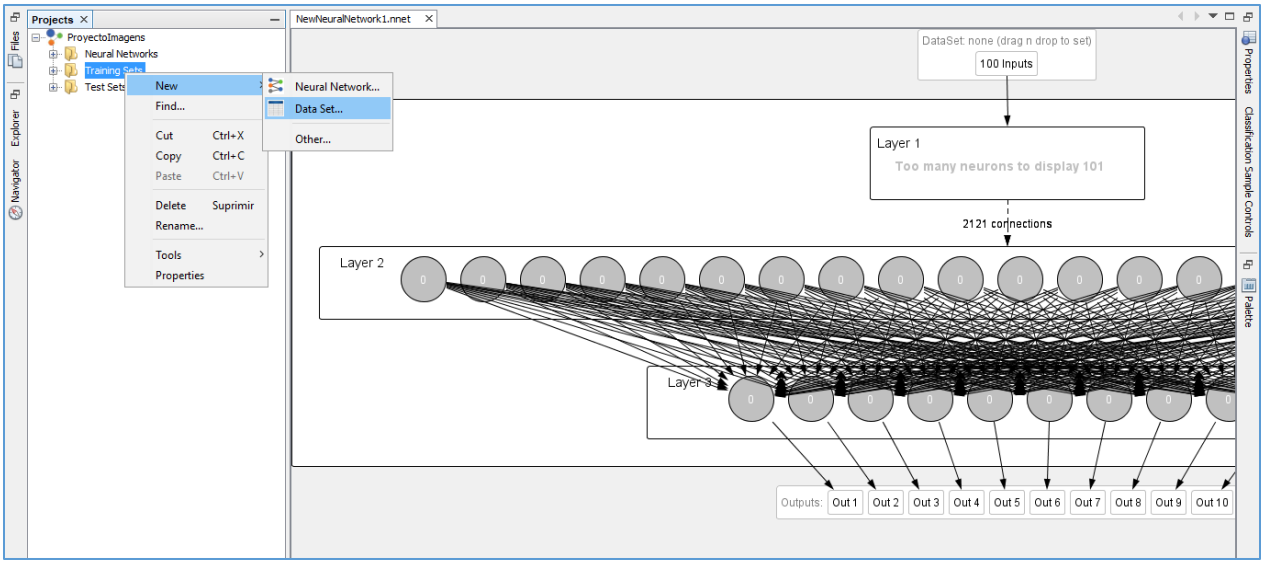

Figura 46: Crear la data para la red.

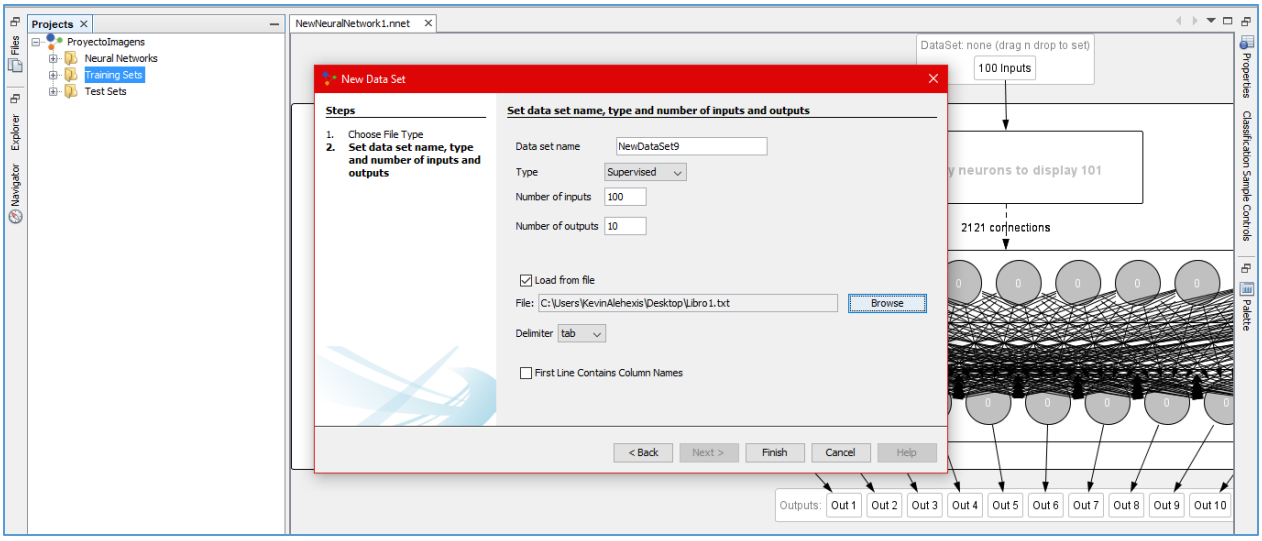

Figura 47: Ingresando los datos de data de la red.

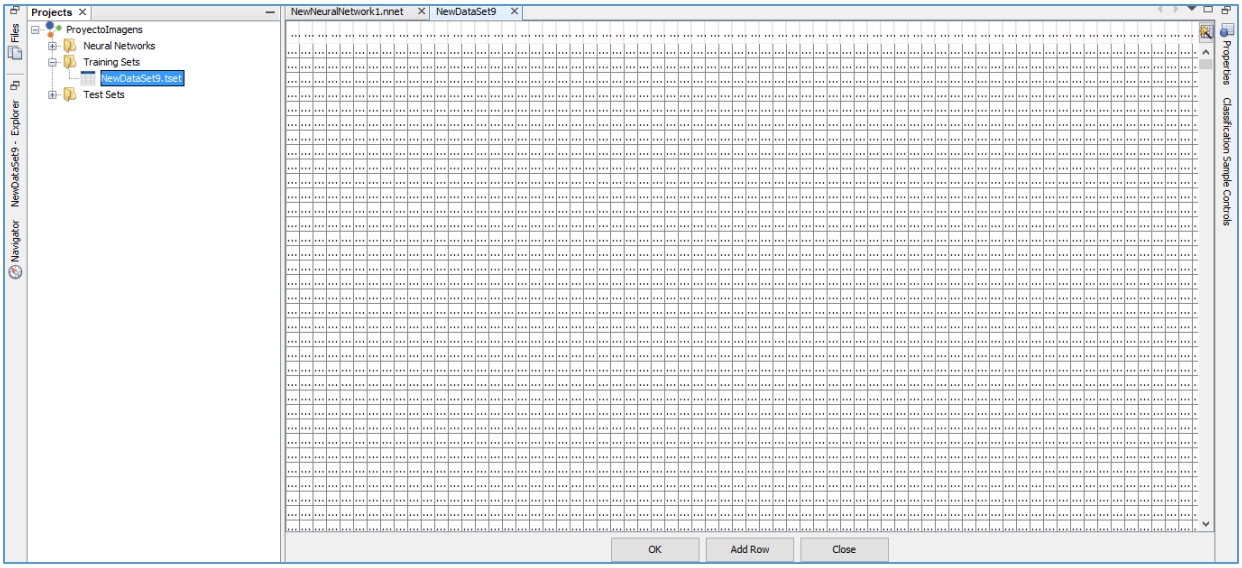

Figura 48: Formato de la data de la red.

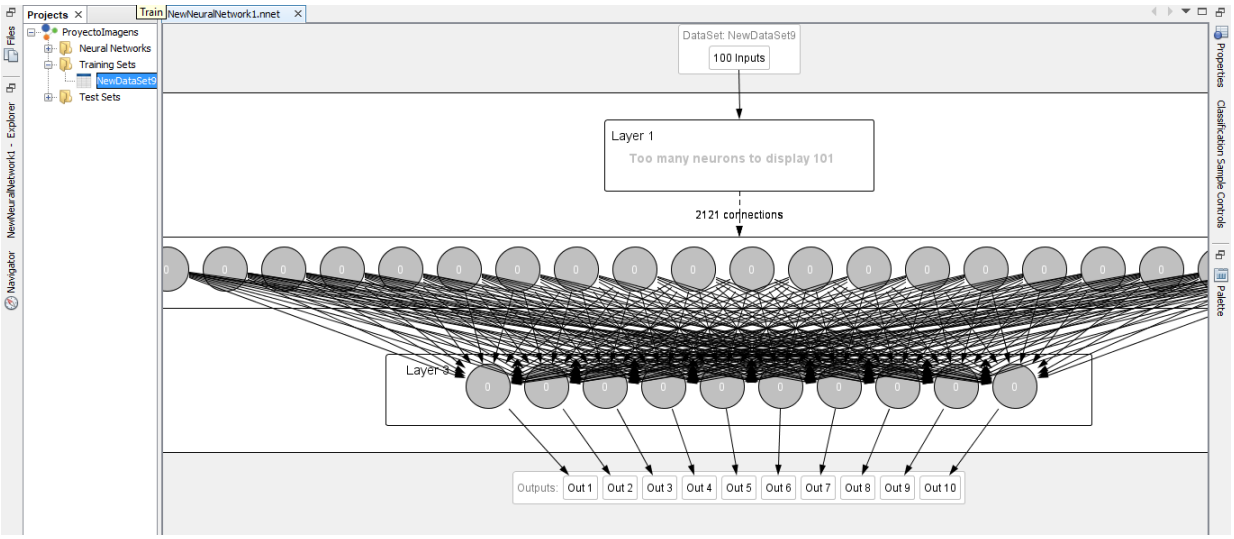

Figura 49:Selecionamos los datos para la red y le damos train.

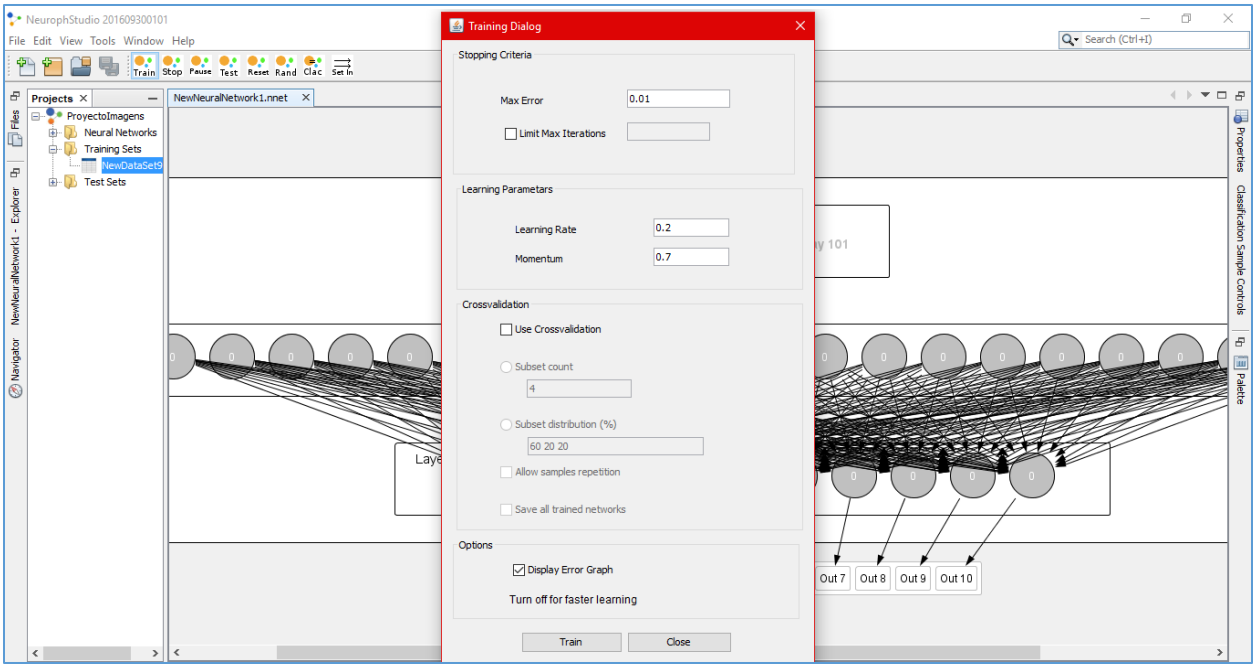

Figura 50: Ingresamos el cantidad maxima de error y ejecutamos.

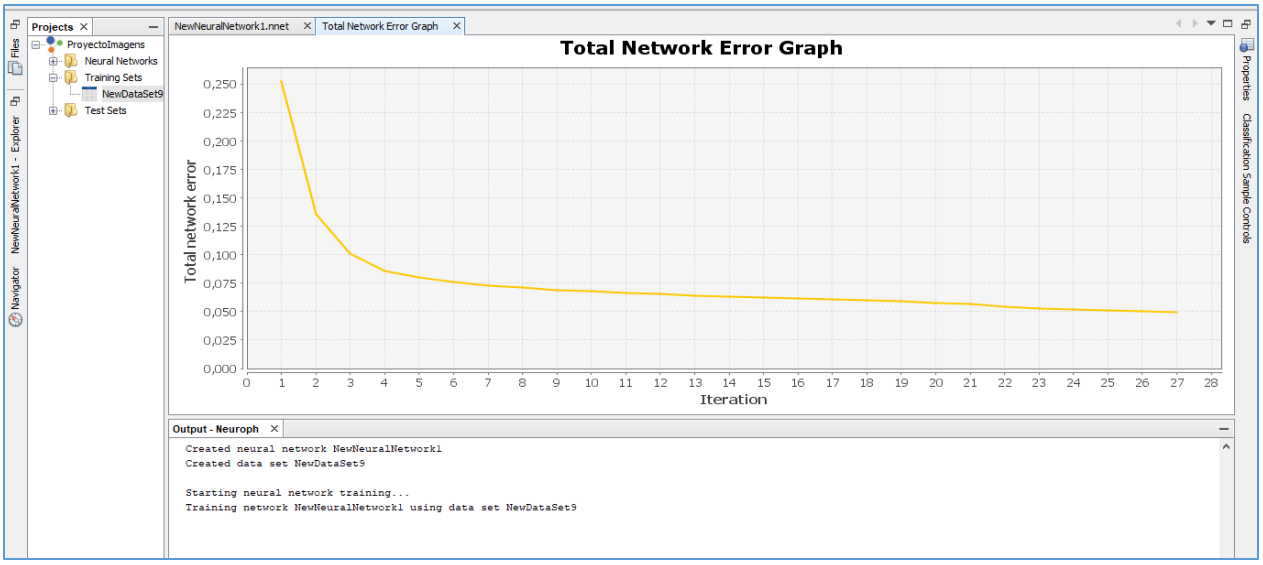

Figura 51: Ejecución de la interacción con la tasa de errort

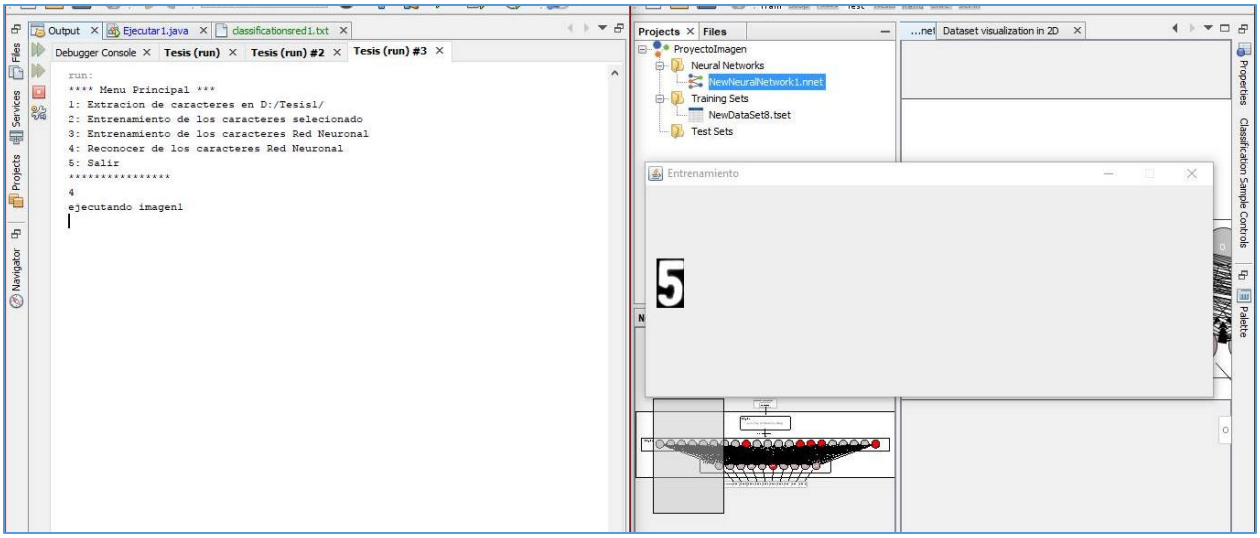

Figura 52: Seleccionamos una imagen y nos saque su contenido.

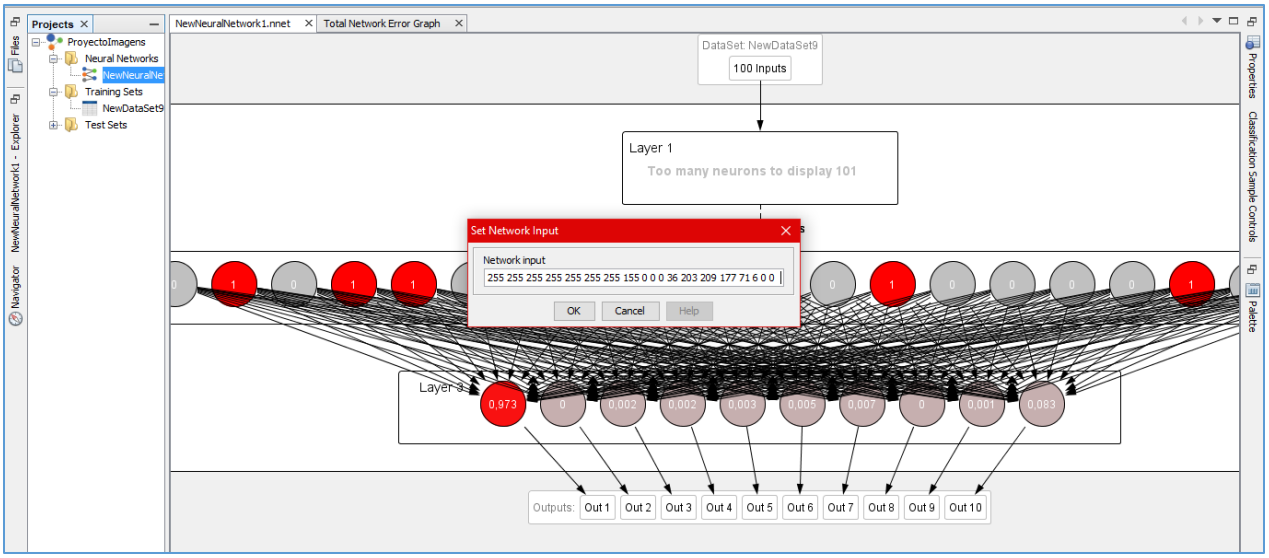

Figura 53: Ingresamos los datos de un digito segmentado para ver sus resultados.

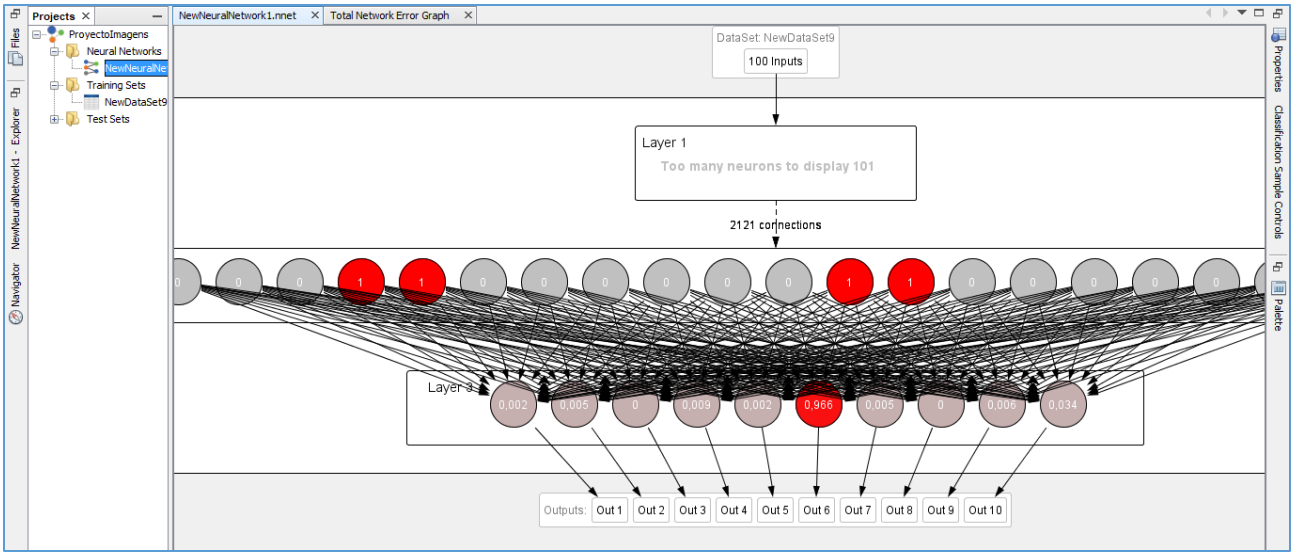

Figura 54: Resultado del inserción de los datos de la imagen.t

Ahora ejecutamos en nuestro proyecto de java haciendo la conexión con el archivo de Neuroph en que nos estos resultados al subir una imagen completa del medidor.

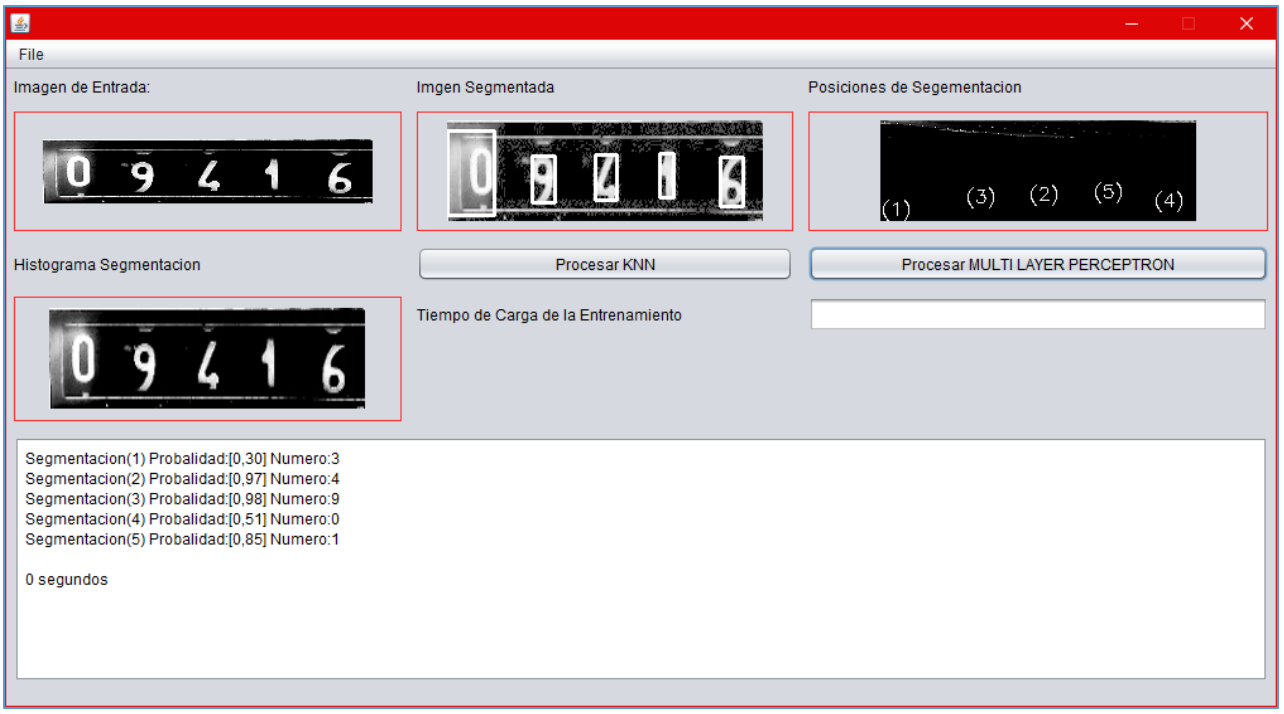

Figura 55: Resultados con la red neuronal

CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

Basado a los resultados presentados en el enfoque de realizar la identificación automática de caracteres en imágenes digitales de consumo de energía eléctrica, se han concluido:

- Para determinar los métodos tenemos que tener en cuenta que algunas investigaciones detallan que los algoritmos son como métodos y otros no, por lo que solo nos hemos enfocado que los métodos son secuencias como se hará la investigación.
- Para la elaboración de capturas es muy recomendable para este tema de tesis usar una secuencia en una creación de una aplicación, porque hacemos un marco de toma para extraer de toda la imagen solamente el consumo del medidor y también que al momento de hacer el recorte es muy recomendable hacer para los medidores mecánicos la conversión de gris en la imagen recortada para reducir los datos de la imagen para luego usarlo en el pre procesamiento.
- Durante de la ejecución de los métodos se concluyó que el uso del algoritmo Otsu es muy recomendable para la enfocar en los bordes de las imágenes al momento de hacer la extracción de contornos ya que tiene un 61.5% de la extracción de los dígitos del consumo de energía eléctrica.
- Tras los resultados hemos tenido un 80% de reconocimiento en las imágenes en la utilización de un clasificador KNN, dando conocer que en la parte del entrenamiento se realizó con un total de 3000 imágenes.
- También detallar que la utilización de la red neuronal ha tenido otros resultados con un 72.7% de reconocimiento, al utilizar 3000 imágenes como data para el entrenamiento y tras la ejecución se vio el conocer que algunos caracteres como 0 y 6 tenía una variación de la precisión de sus resultados.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

6.2. Recomendaciones

- Se recomienda para la utilización de tema de tesis basado a medidores utilizar otros protocolos de tomas fotográficas que se presenta en otras investigaciones, si se hace en tiempo real para las tomas.
- Para el entrenamiento se recomienda en utilizar otros algoritmos para tener una diferencia en las investigaciones futuras.
- En proyectos futuros ya se tendrá una data de la clasificación basada a los modelos de dígitos de consumo de medidores para la utilización en otros proyectos de investigación y la automatización automática de un aplicativo móvil.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

7. Anexos.

En estos anexos se detallan como se configuro las librerías y codificación del aplicativo:

- Configuración de la librería OpenCV tanto móvil como de escritorio.
- Codificación de la aplicación de plan de capturas.
- Codificación del aplicativo para el pre procesamiento.
- Codificación del aplicativo para el entrenamiento.
- Codificación del aplicativo para ejecutar los resultados.

7.1. Configuración de la librería OpenCV tanto móvil como de escritor.

7.1.1. Configuración de OpenCV SDK en aplicación móvil.

Para la configuración de OpenCV en el aplicativo móvil para el plan de capturas vamos a crear la aplicación con el IDE de Android Studio detallando la realización de estos pasos:

- Crear el aplicativo móvil con el IDE de Android studio.
  Para este paso vamos a crear un aplicativo móvil abriendo el IDE de Android Studio.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Figura 56: Plataforma del aplicativo IDE Android Studio 2.3.3

Al seleccionar el “Start a new Android Studio project” se mostrará este cuadro donde hace la especificación del nombre del proyecto, el nombre del dominio del proyecto o mejor conocido el nombre de los paquetes donde estará el proyecto y la ubicación del proyecto donde se va almacenar.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

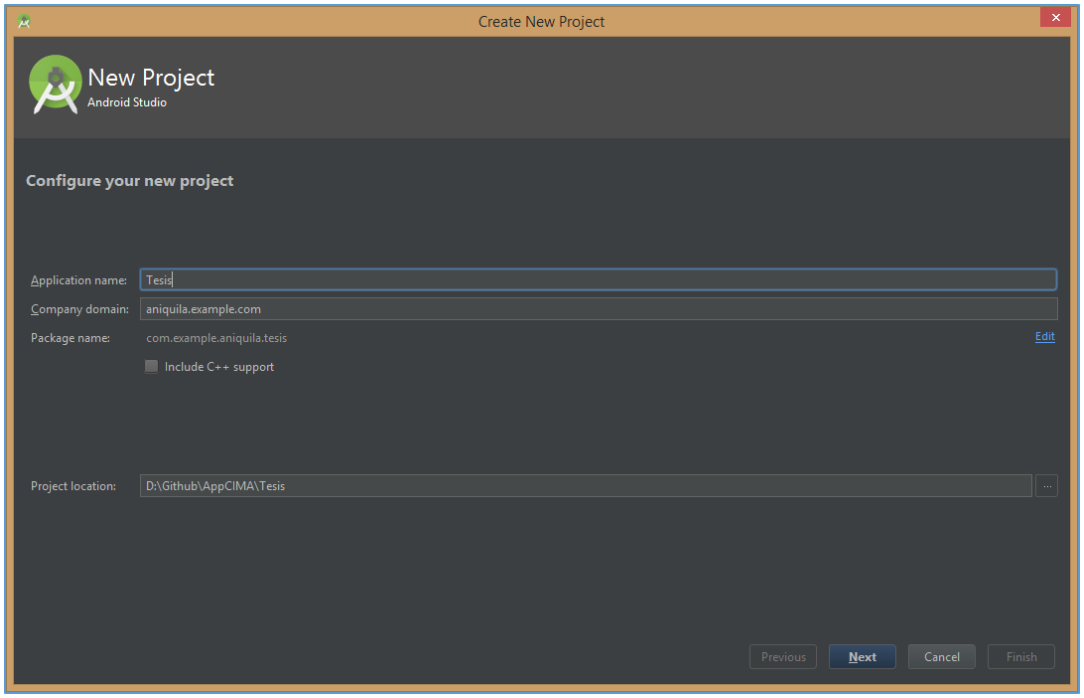

Figura 57: Especificaciones del nombre y paquetería del proyecto.

Basado a la imagen 29 vamos a dar "next" o continuar, luego de eso se mostrará la especificación del aplicativo en que aquí se detallara como se va especificar el proyecto como: Móvil o Tablet, Wear, Tv, Android Auto; donde nosotros solamente vamos a utilizar el Móvil o Tablet.

Al seleccionar Móvil y Tablet vamos a especificar desde que versión mínima de Android será el aplicativo

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

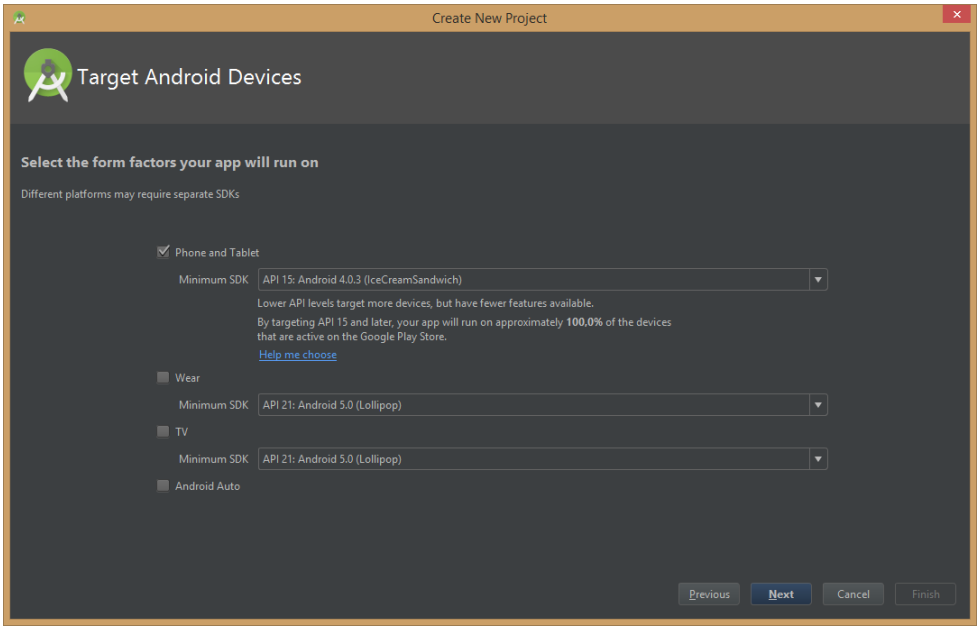

Figura 58:Seleccionamos el modo de la app.

Luego al dar continuar se mostrará una ventana con el modelo del primer diseño del aplicativo, por lo que dejaremos por defecto ese diseño.

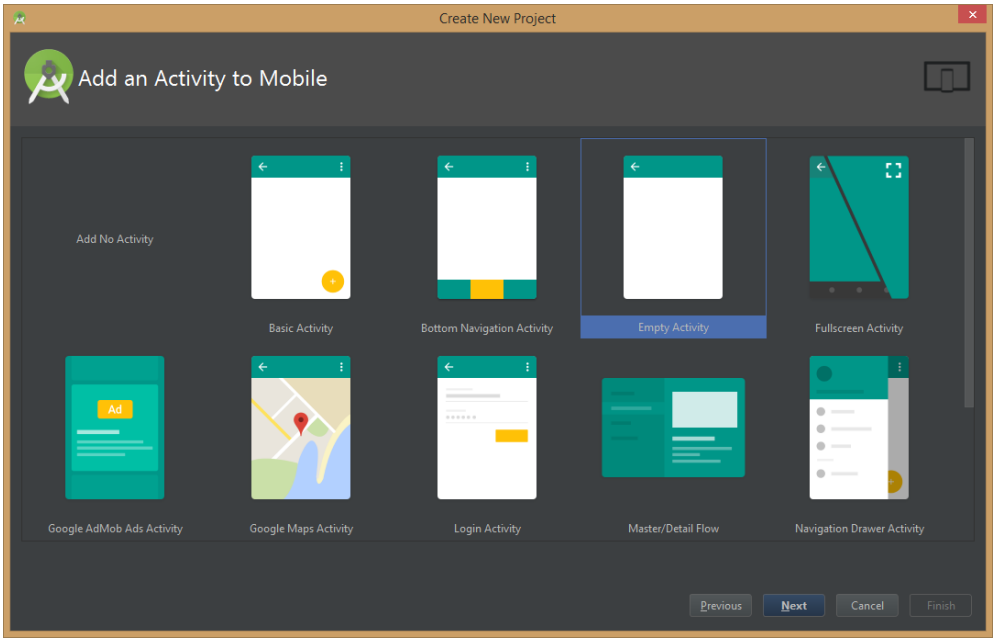

Figura 59: Modelo del primer diseño del aplicativo.

Despues de dar el formato del diseño nos abrira una nueva

ventana cuando le damos a continuar, aquí se detalla el nombre del primer actividad o la clase ejecutadora del proyecto.Teniendo en cuenta que esa actividad es la parte principal donde se ejecutara las primeros eventos de la aplicación.

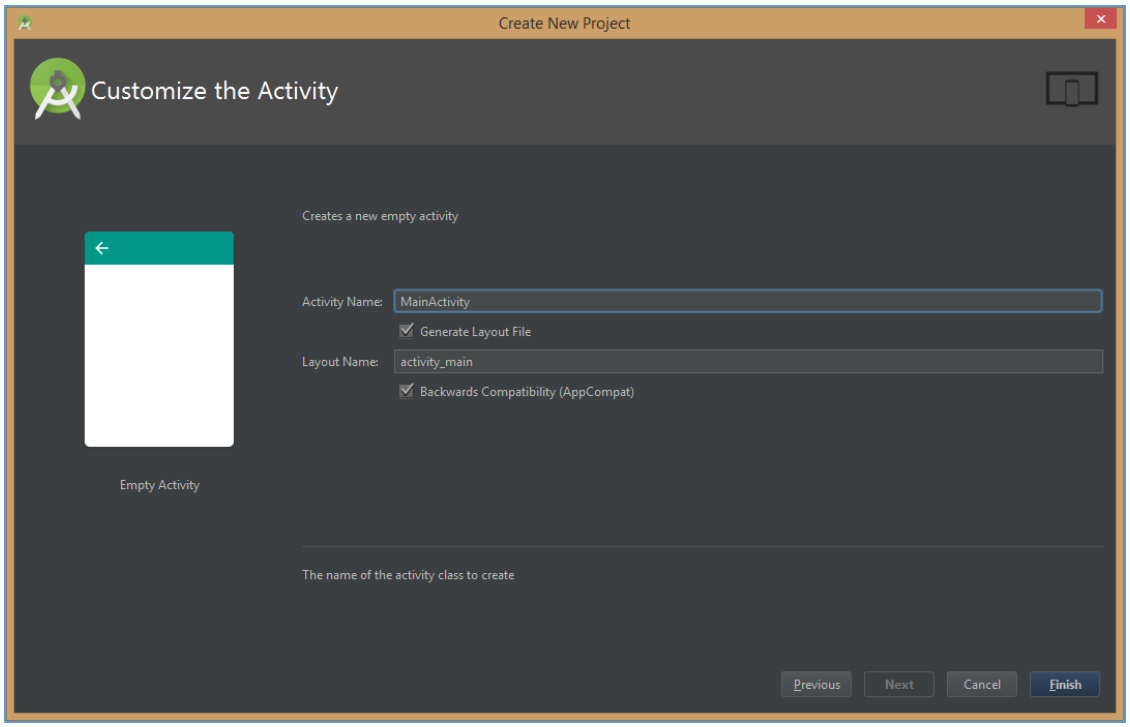

Figura 60: Modelo de la primera actividad del proyecto.

Luego de darle finalizar en de la ventana anterior se ejecutara la carga de los datos de la aplicación y la ejecucion interna del IDE Android Studio.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

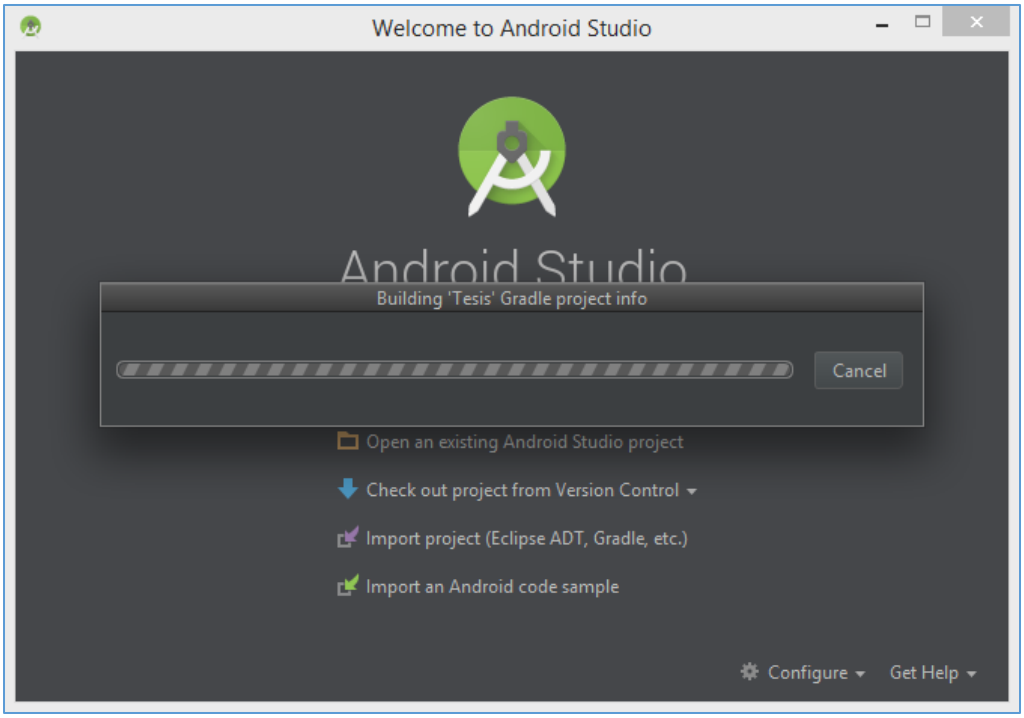

Figura 61: Cargando los datos del proyecto.

Luego de ejecutar la creacion del proyecto se mostrar la platafoma completa del IDE de Android Studio.

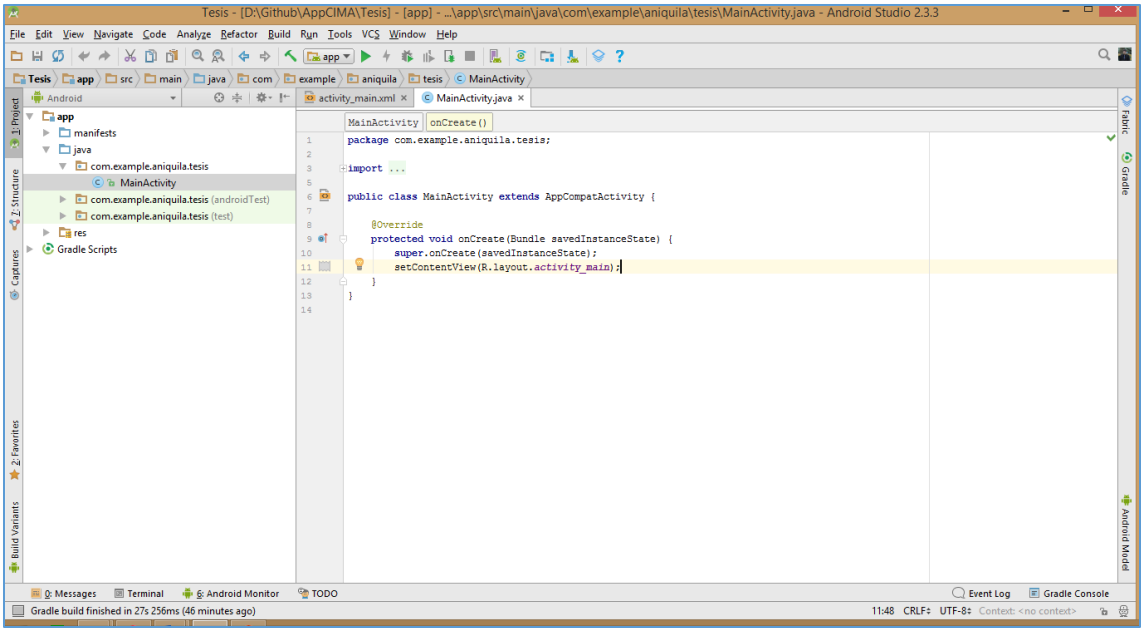

Figura 62: Plataforma de Android Studio.

- Descarga de la librería de OpenCV el aplicativo móvil.

  Para descargar la librería de OpenCV debemos entrar a la página oficial de OpenCV.

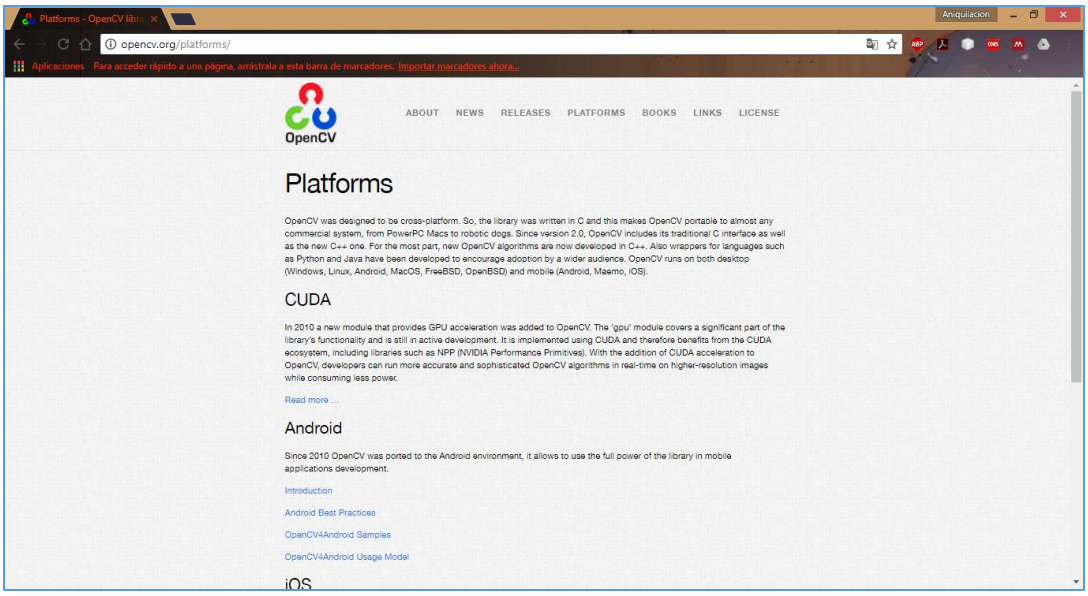

Figura 63: Página Oficial de OpenCV.

En la página oficial de OpenCV vamos a entrar las plataformas que soporta OpenCV y seleccionamos Android para seleccionarlo en la opción “introducción”, en que allí vamos a descargar la librería.

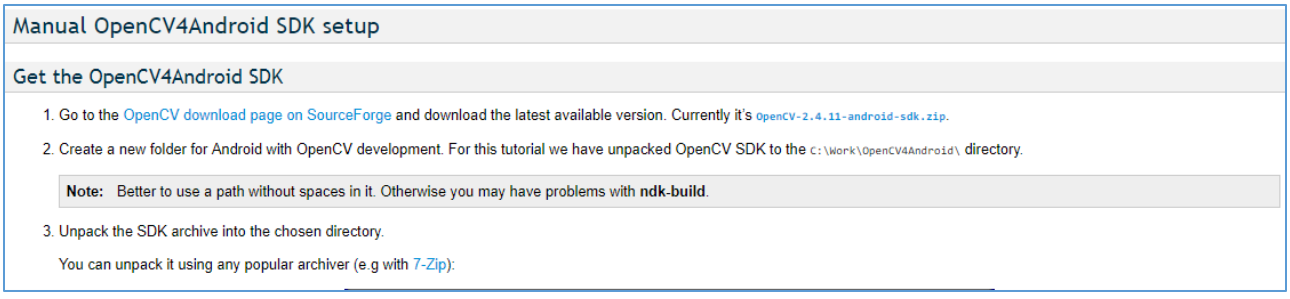

Figura 64: Manual de instalación de OpenCV en Android.

En la opción marcada “OpenCV download page on SourceForge” allí está el repositorio de los archivos para el OpenCV SDK para el Android Studio.

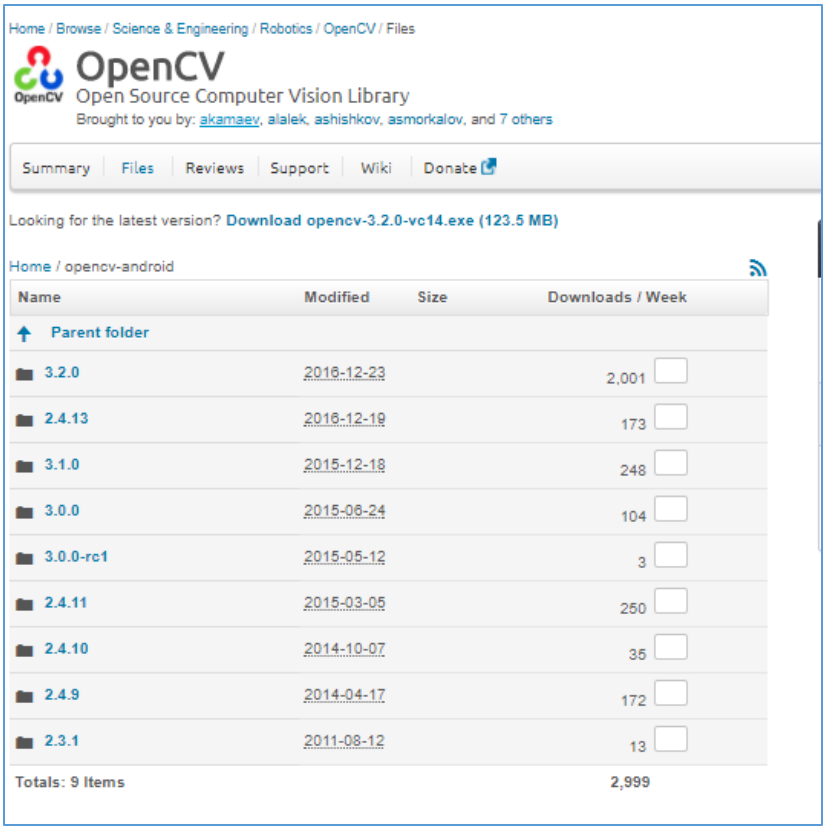

Figura 65: Repositorio de OpenCV para Android.

Basado a eso vamos a descargar la última utilización de OpenCV.

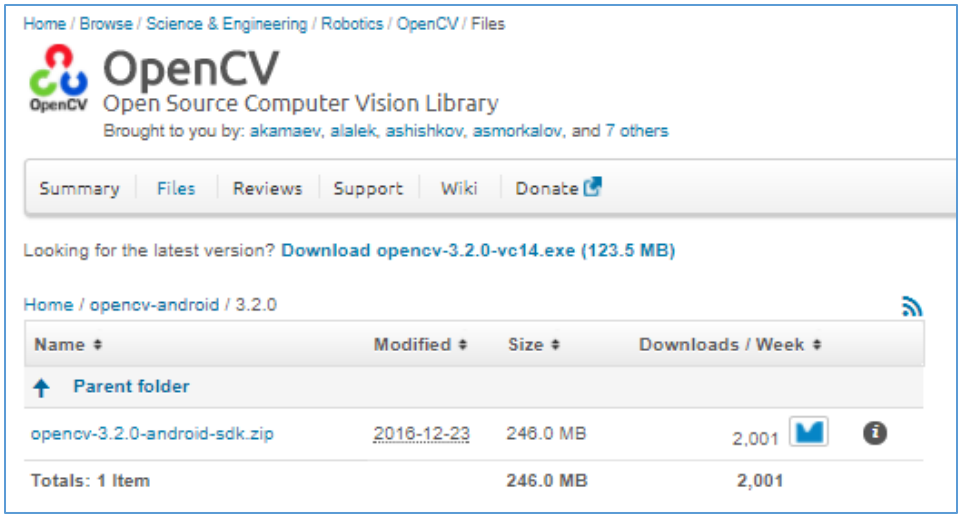

Figura 66: Archivos del OpenCV de Android.

Al descargar este paquete vamos a descomprimir el

archivo y situarlo en la carpeta principal del sistema de la computadora.

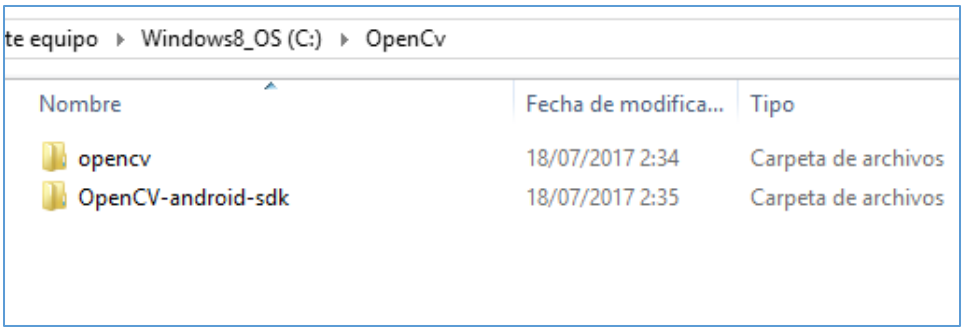

Figura 67: Archivos descomprimidos de OpenCV SDK.

- Importación del módulo OpenCV SDK al IDE Android Studio.

  Para la importación del módulo debemos seleccionar del Android studio la carpeta de OpenCV SDK.

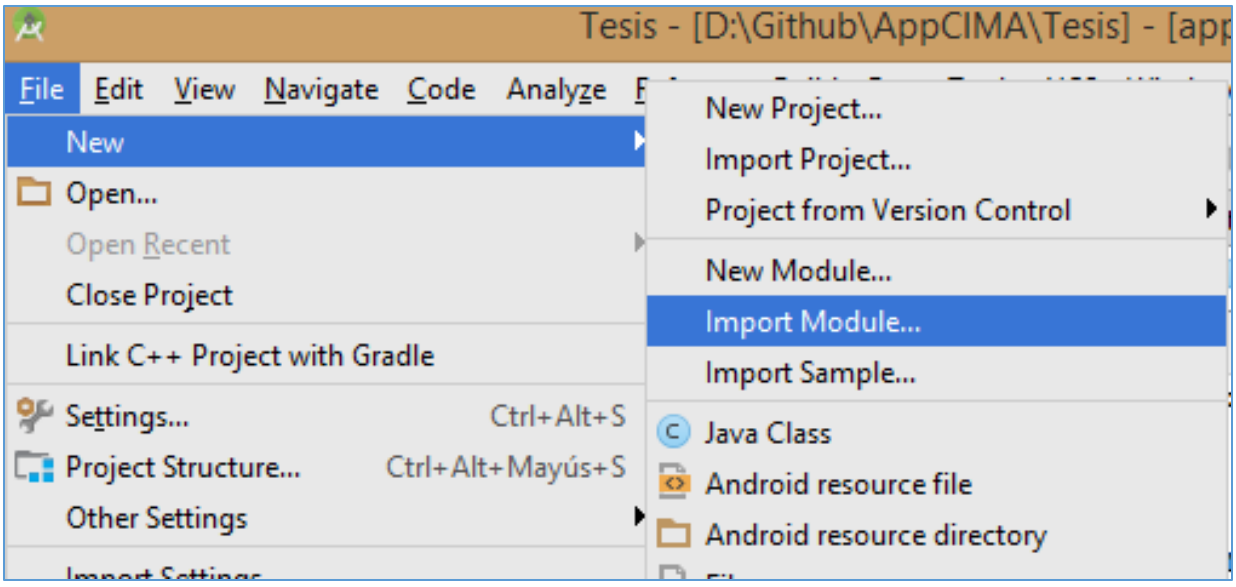

Figura 68: Seccionar la opción de importar modulo.

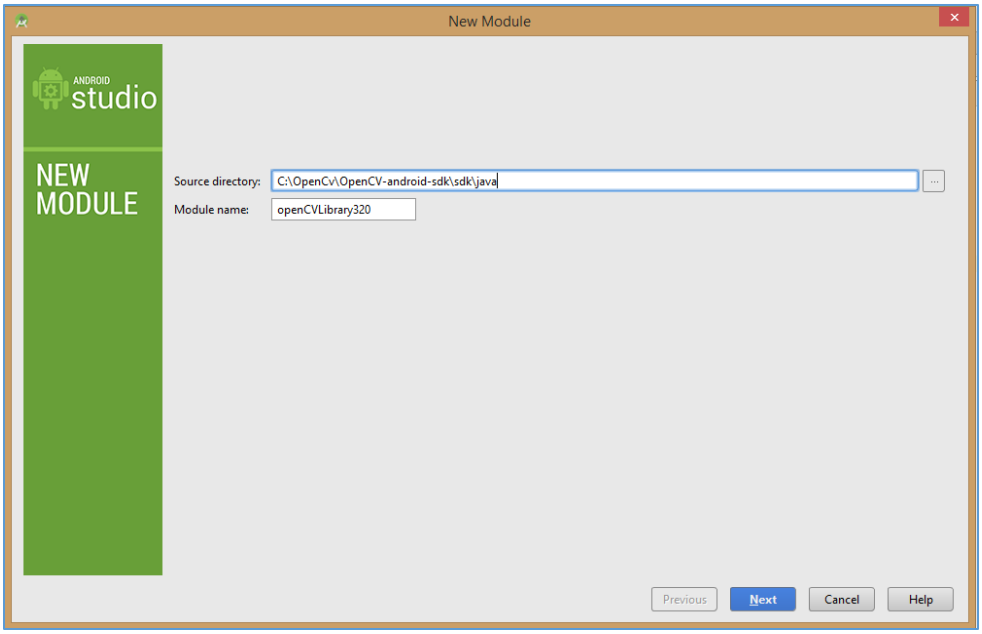

Figura 69: Seleccionamos el OpenCV al Android Studio.

Al dar siguiente en la ventana nos dará una recomendación de la importación para luego finalizar la importación.

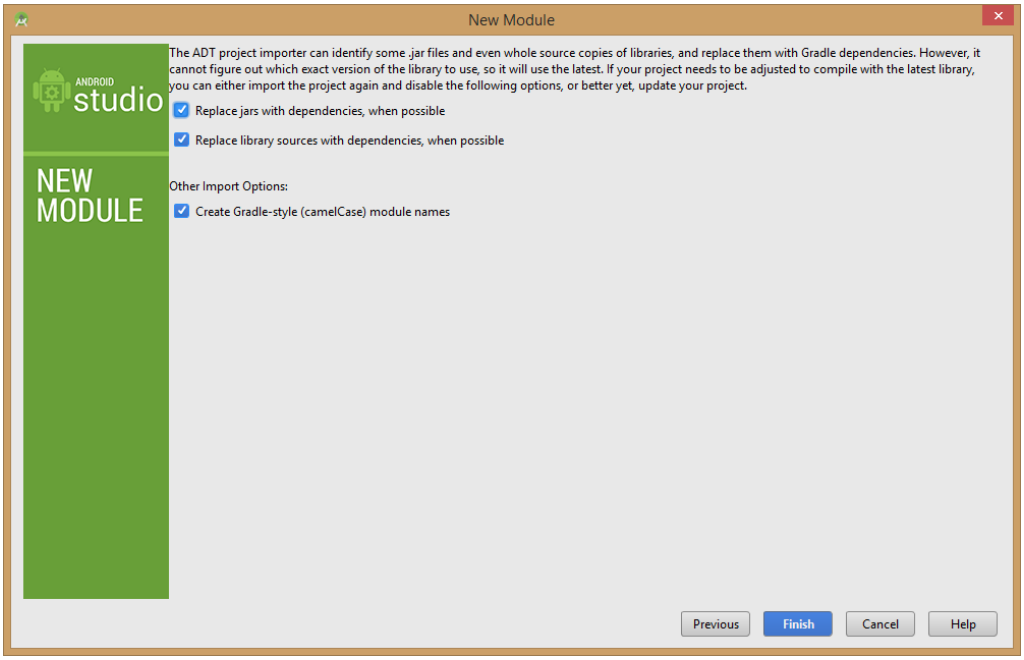

Figura 70: Referencias de la importación del módulo.

Luego ya importado debemos configurar algunos campos que viene del OpenCV que se adapte al proyecto creado.

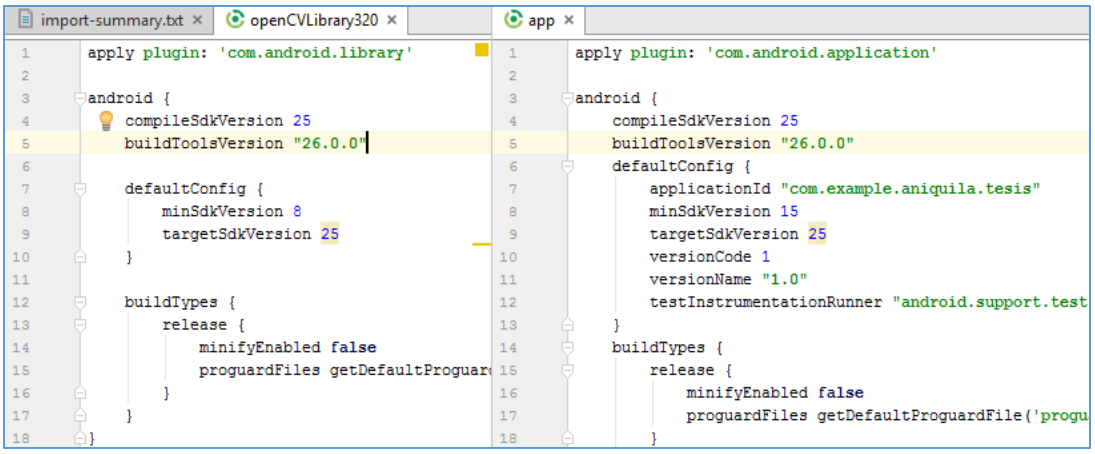

Figura 71: Configuración del Gradle del OpenCV SDK.

Cuando se actualiza el gradle después de modificar el gradle del módulo de OpenCV SDK vamos a actualizar todo el proyecto para luego llamar en el gradle del proyecto el módulo de OpenCV SDK.

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:25.3.1'
    compile 'com.android.support.constraint:constraint-layout:1.0.2'
    compile project(path: ':openCVLibrary320')
}
```

Figura 72: Modificación del gradle del proyecto principal.

Luego de importar el módulo OpenCV SDK vamos a importar los JNI del módulo al proyecto.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

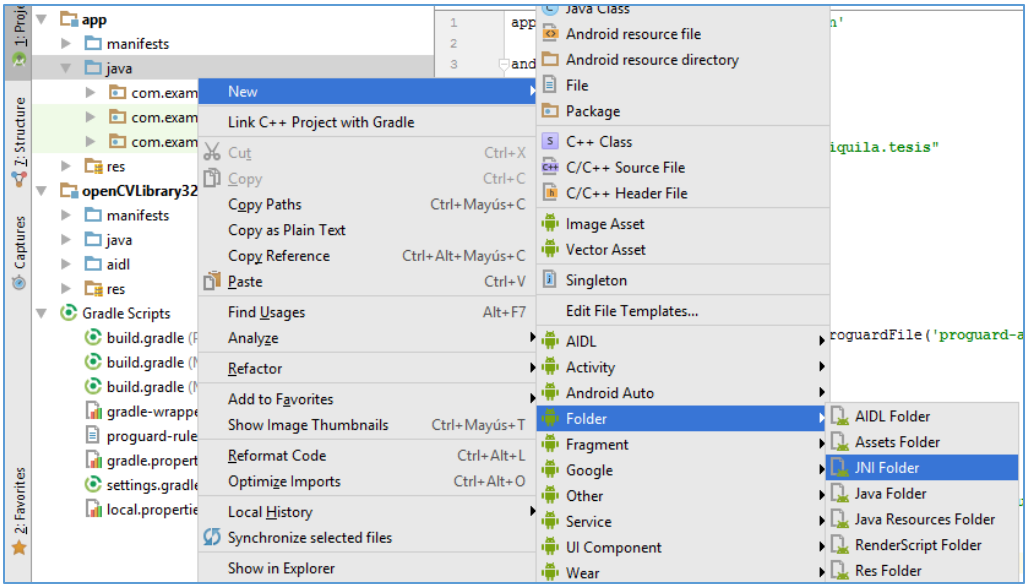

Figura 73: Seleccionamos el JNI Folder para importar algunos paquetes del SDK.

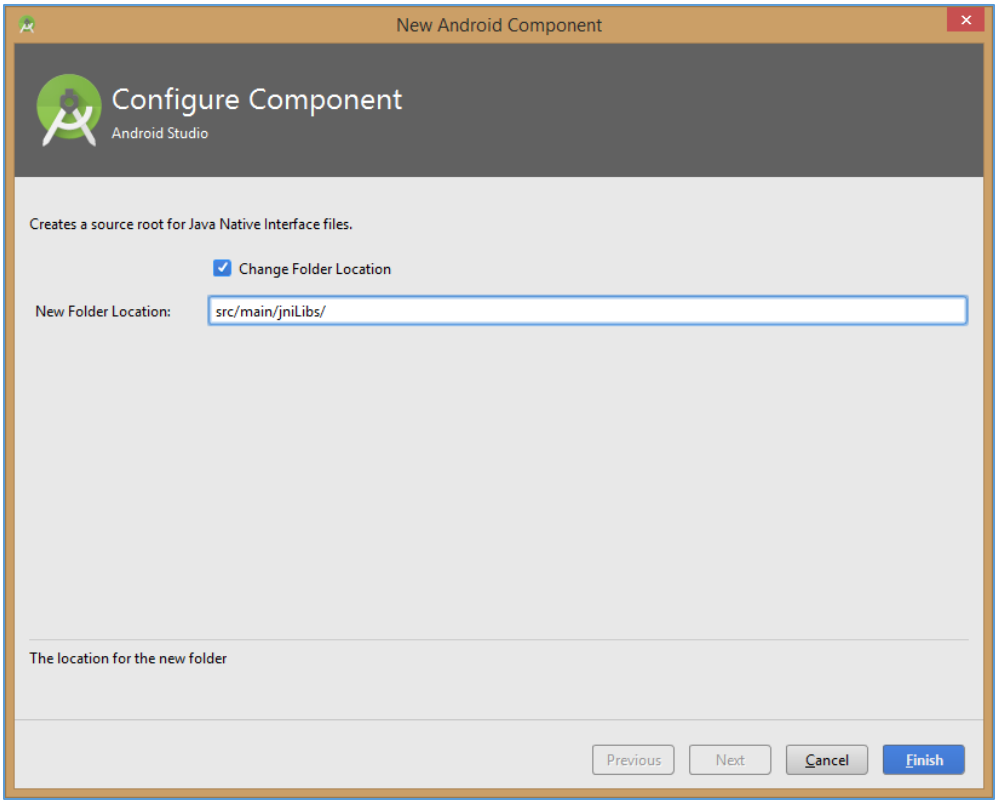

Figura 74: Configurar el JNI del proyecto.

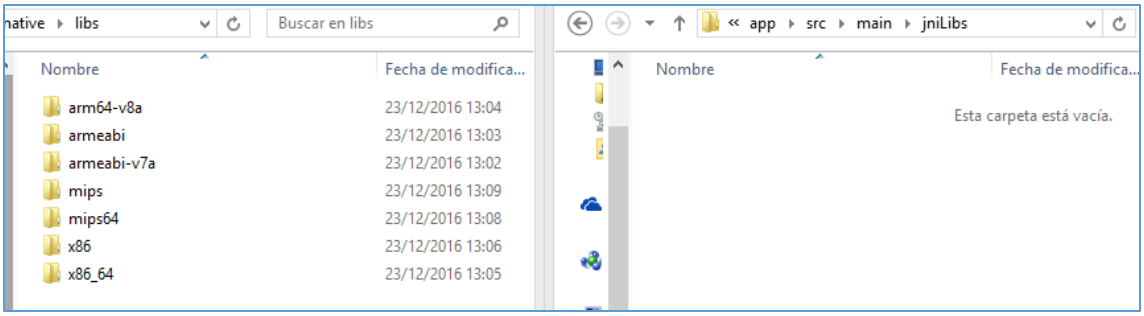

Figura 75: Archivos de JNI de OpenCV SDK al JNI del proyecto.

Vamos a copiar y pegar todos los archivos de libs al JNI libs en el proyecto de Android Studio se vera de esta forma.

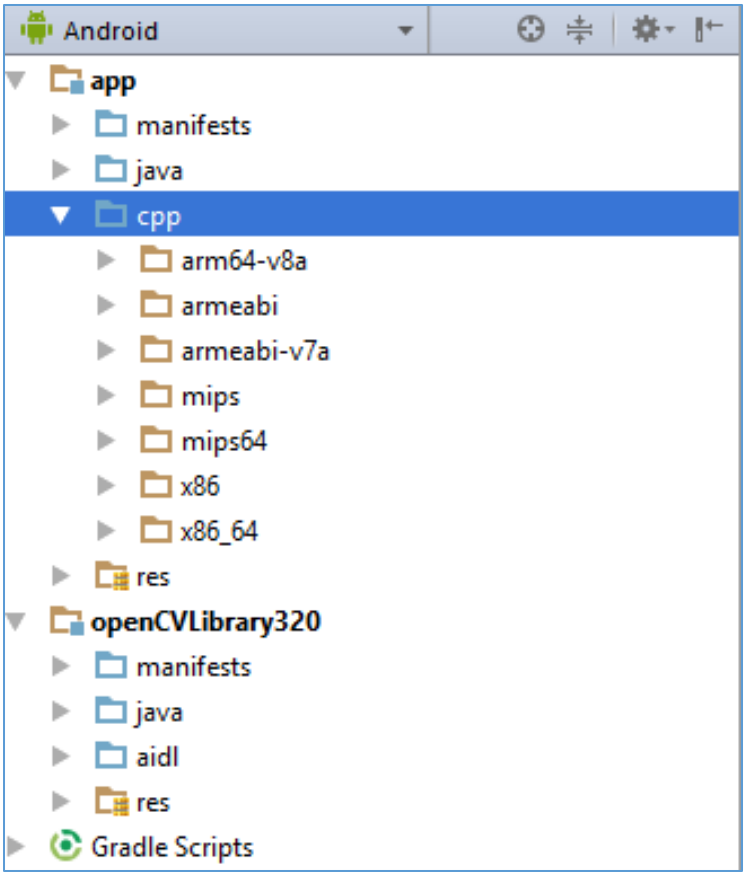

Figura 76: Archivos importados del JNILibs.

- Resultados de la importación.

  Pasando toda la configuración vamos a llamar una clase principal del proyecto vamos hacer una validación si se importó los archivos o no.

```java
package com.example.aniquila.tesis;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        if (!OpenCVLoader.initDebug()) {
            Log.e("json", "no cargo los paquetes OpenCV_SDK");
            Toast.makeText(this, "no cargo los paquetes OpenCV_SDK", Toast.LENGTH_LONG).show();
        } else {
            Log.e("json", "cargo los paquetes OpenCV_SDK");
            Toast.makeText(this, "cargo los paquetes OpenCV_SDK", Toast.LENGTH_LONG).show();
        }
    }
}
```

Figura 77: Codificación de validación de importación.

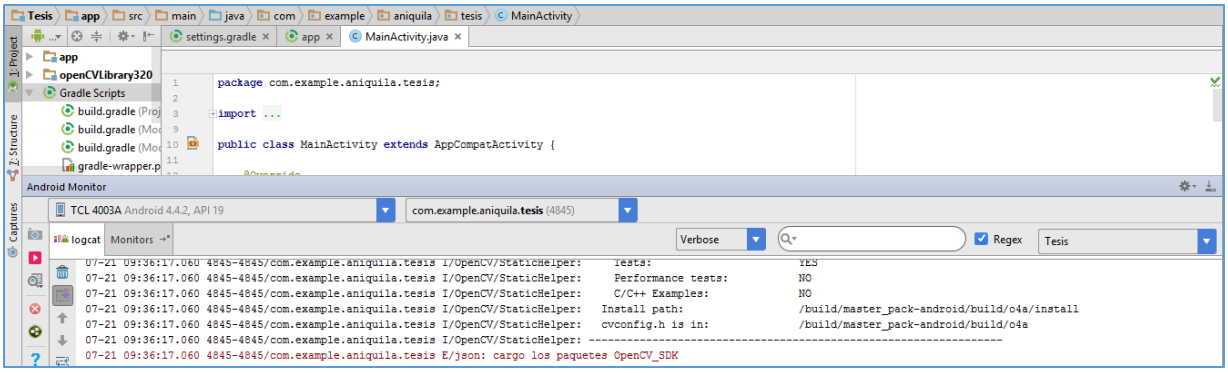

Figura 78: Resultado de la importación de la librería.

7.1.2. Configuración de OpenCV en aplicación de escritorio.

- Crear el aplicativo con el IDE de Netbeans.

  De la plataforma IDE Netbeans se hará una aplicación escritorio con el lenguaje de programación Java.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

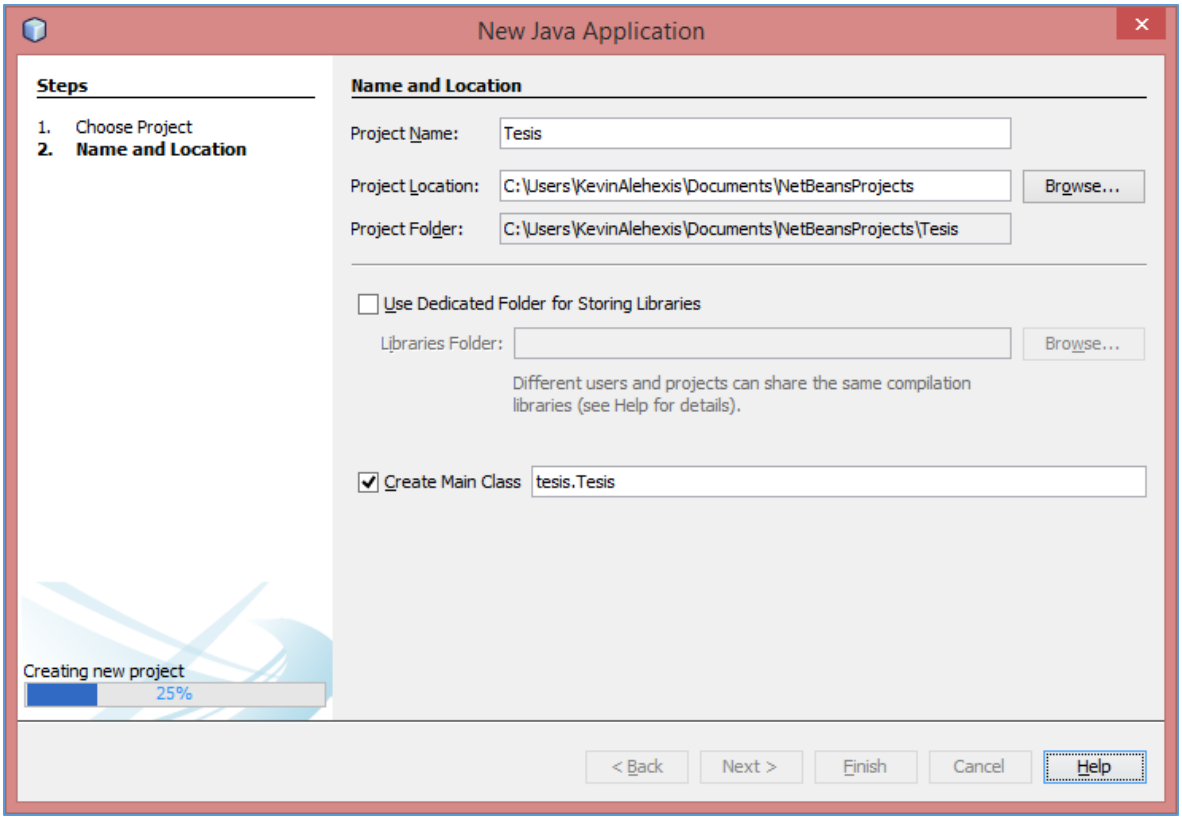

Figura 79: Creación del proyecto escritorio en IDE Netbeans.

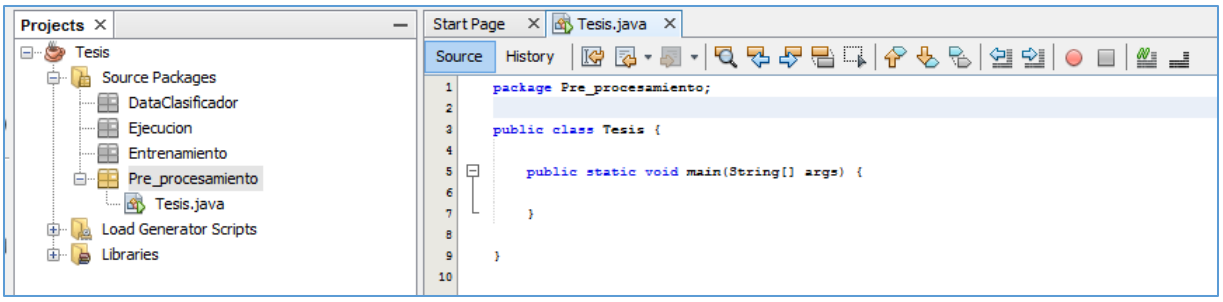

Figura 80: Estructura del proyecto.

- Descarga de la librería de OpenCV para escritorio.
  Para la aplicación escritorio se descargar la librería desde la misma página web del OpenCV en que vamos a seleccionar el ultimo librería que ha sido actualizado por OpenCV.

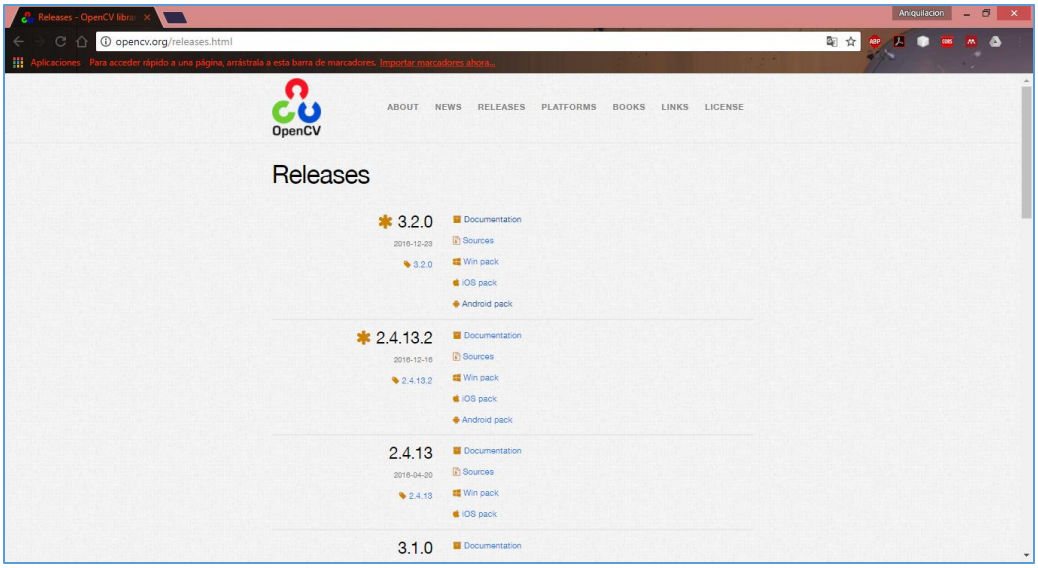

Figura 81: Pagina para descargar el módulo de OpenCV.

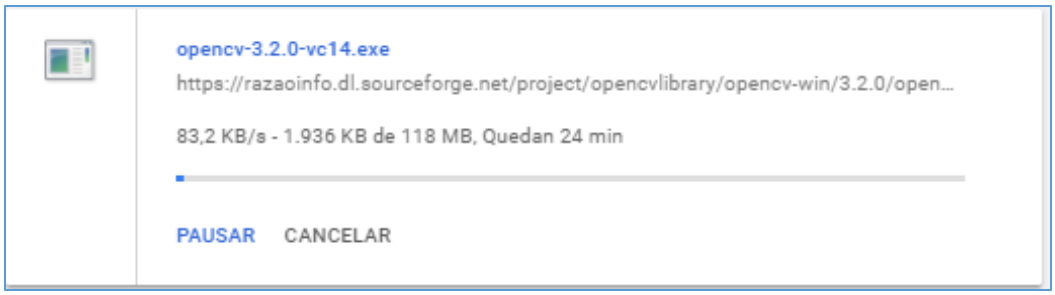

Figura 82: Archivo de instalación de OpenCV.

Después de descargar el modulo vamos a ejecutar y nos pedirá la ubicación de la instalación por lo que se hace una recomendación ubicar en la disco principal de la máquina.

- Importación del módulo OpenCV al IDE de Netbeans.

  Después de guardar en la maquina tienes así este formato de los archivos del OpenCV.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

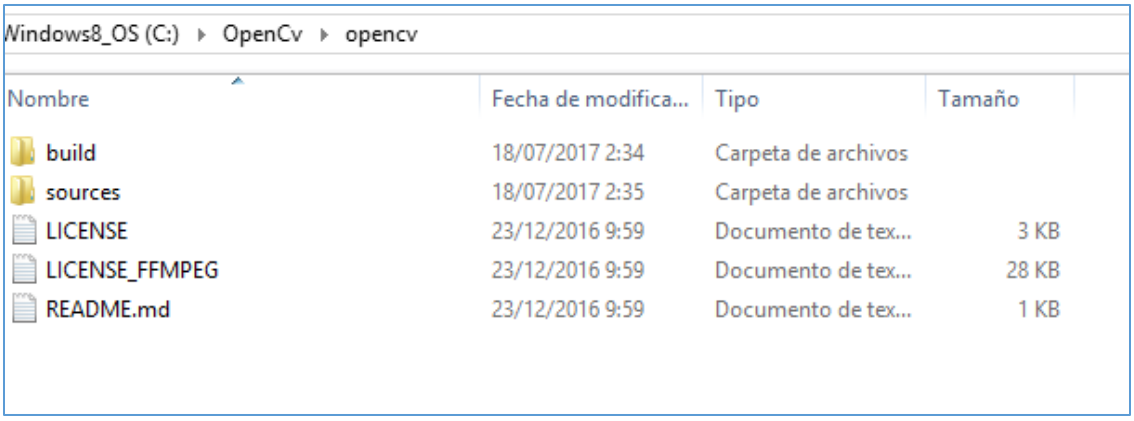

Figura 83: Archivos del OpenCV.

Al tener esos archivos vamos a llamar al IDE de Netbeans para que haga el reconocimiento de los paquetes y clases del OpenCV.

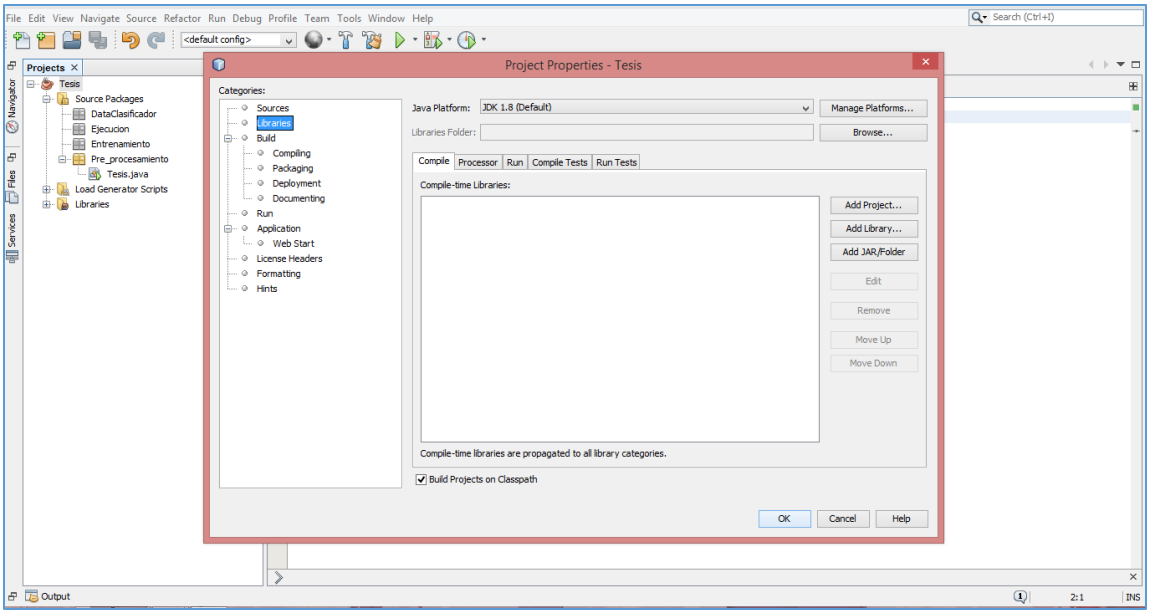

Figura 84: Plataforma IDE Netbeans para llamar el Jar de OpenCV.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

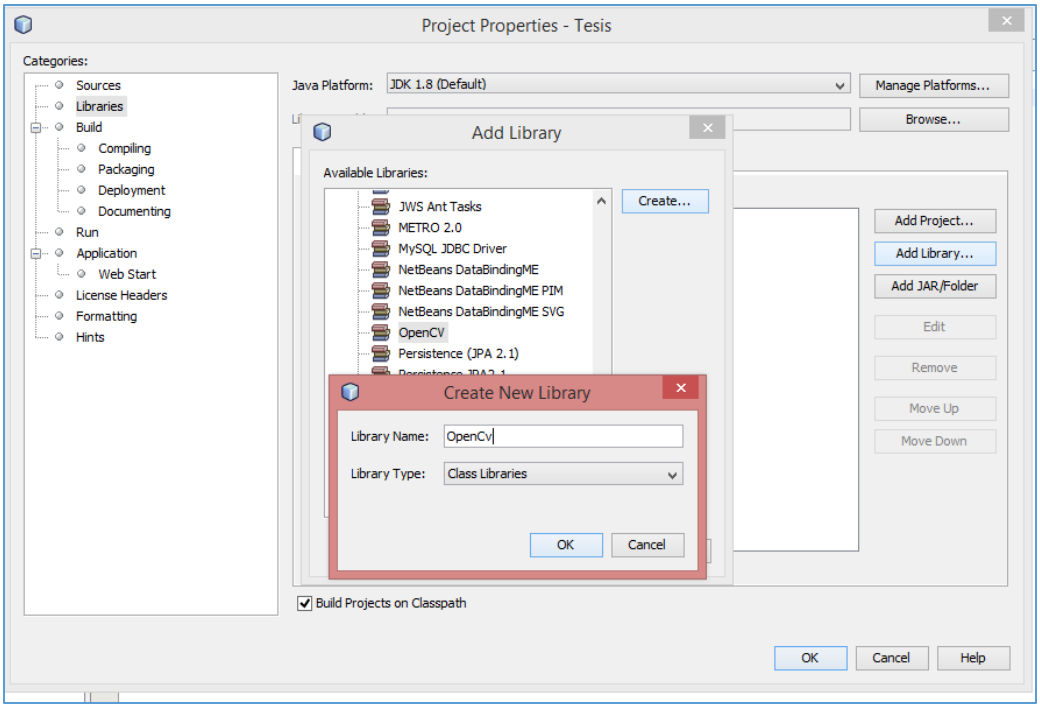

Figura 85: Crear un módulo que solo sea OpenCV.

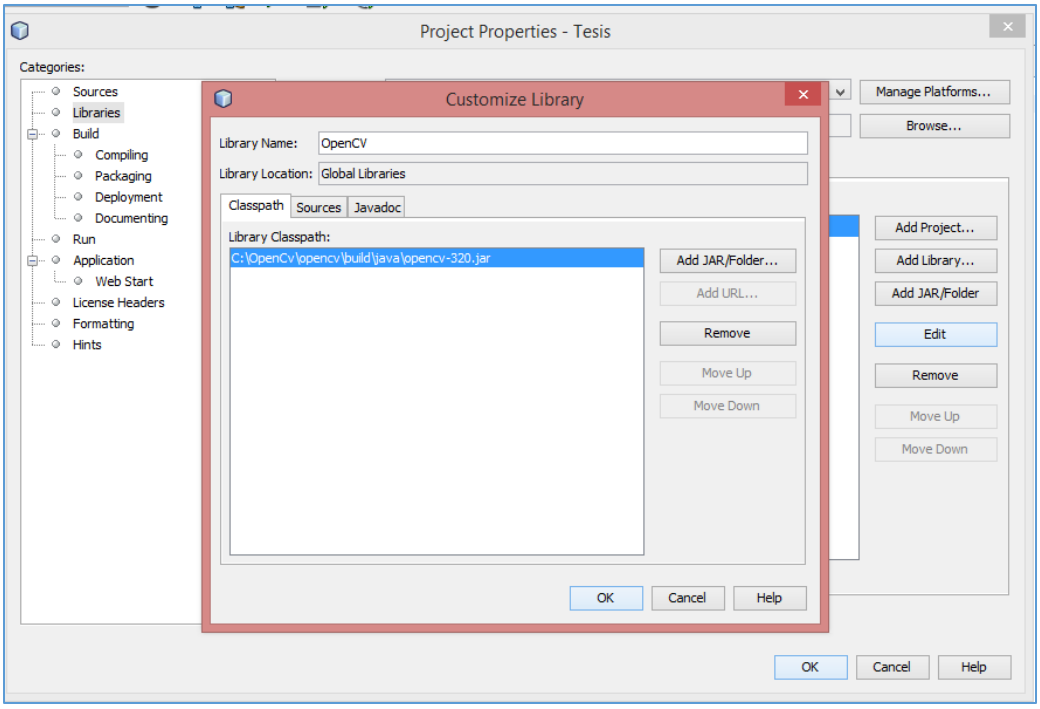

Figura 86: Implementar el Jar del OpenCV al proyecto.

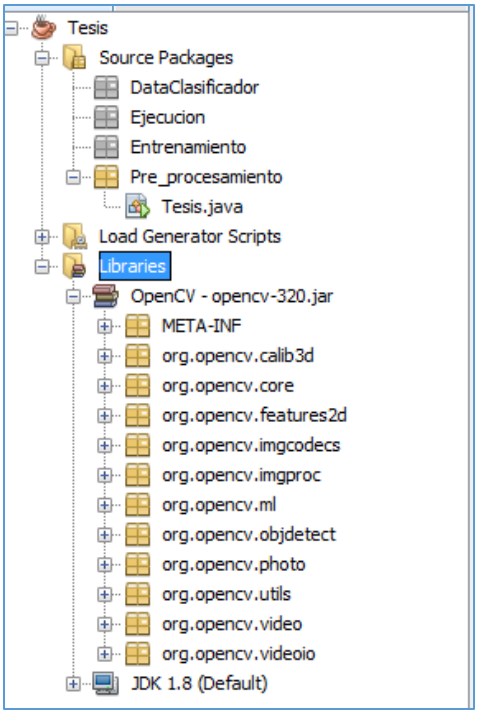

Figura 87: Paquetes del OpenCV.

Después de importar el módulo de OpenCV ahora hay que copiar el archivo "opencv_java320.dll" de la ubicación del "opencv\build\java\(versión de la maquina x64 o x86)" al proyecto principal para el reconocimiento de algunos paquetes del OpenCV.

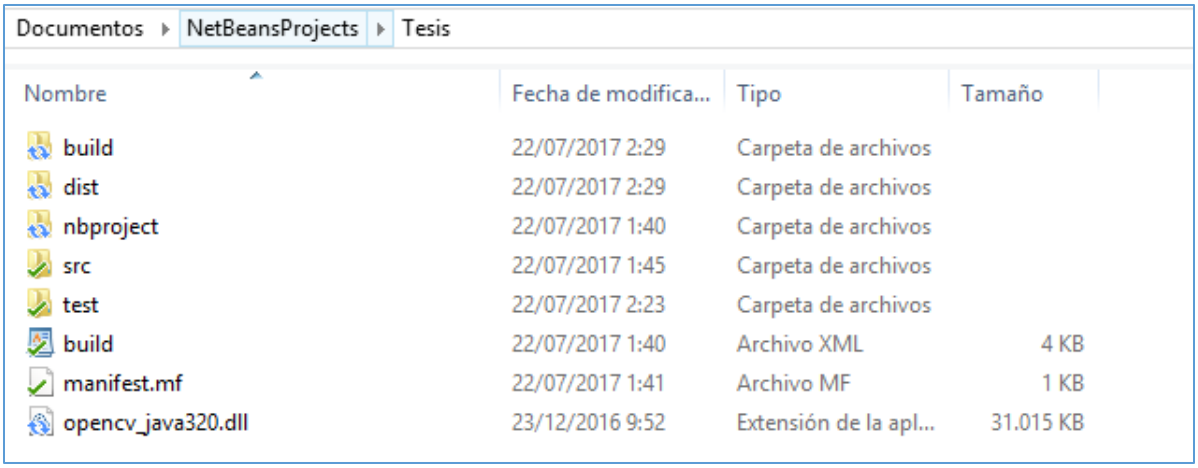

*Figura 88: Importación del *.dll al proyecto.*

- Resultados de la importación.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Vamos a ejecutar un pequeño ejemplo con los contenidos del OpenCV utilizando la clase Core y Mat.

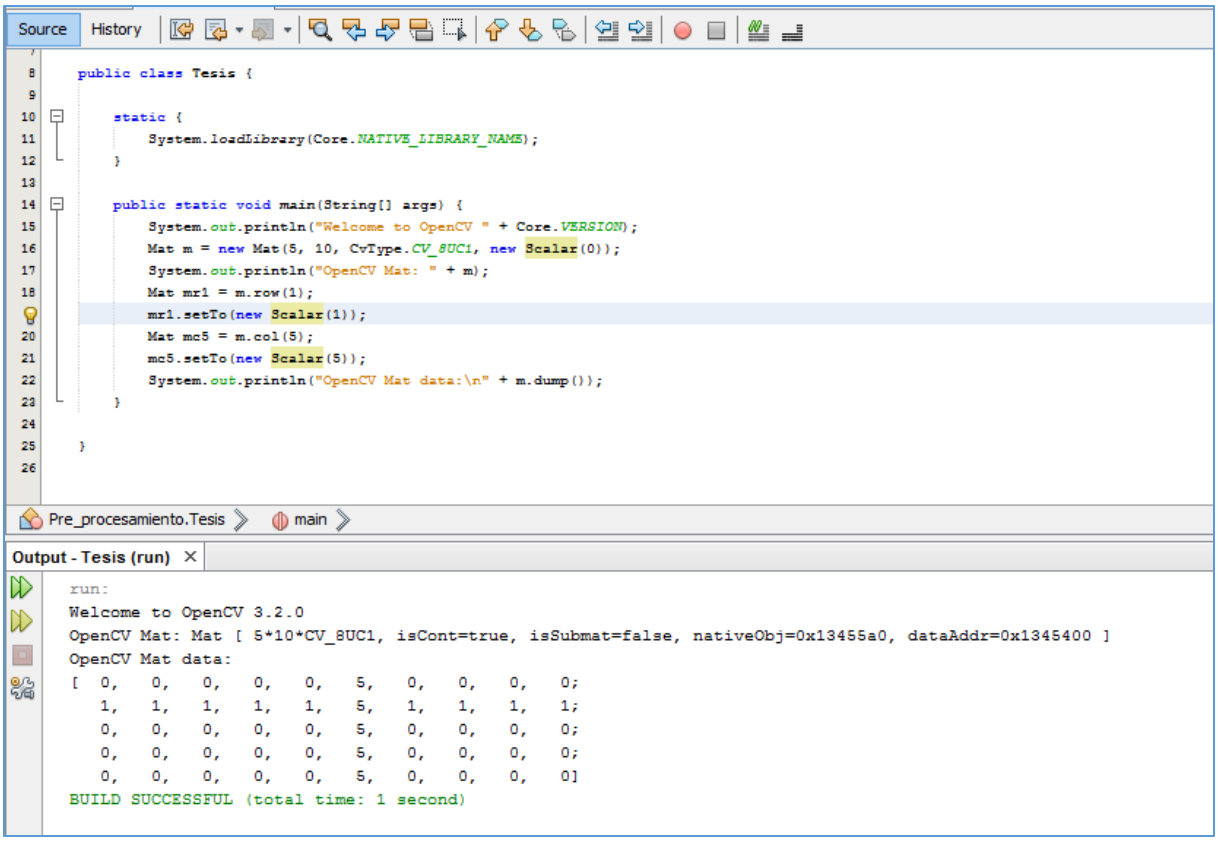

Figura 89: Ejecución de las pruebas de importación del OpenCV.

7.2. Codificación del aplicativo móvil para el plan de capturas.

- **Modificación del AndroidManifest.**

  Para la ejecución de la cámara del aplicativo tenemos que activar los permisos de la cámara y el almacenamiento de los archivos de las tomas fotográficas.

```
<uses-feature android:name="android.hardware.camera" />
<uses-feature android:name="android.hardware.camera.front" />
<uses-feature android:name="android.hardware.camera.autofocus" />
<uses-feature android:name="android.hardware.camera.front.autofocus" />

<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Figura 90: Permisos para la cámara y almacenamiento.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

- **Crear una clase para la visualización de la cámara.**

  Se realizará una clase que contenga el visualizador de la cámara junto con adaptación de aumento de zoom de la cámara para tener el mejor posicionamiento del consumo del medidor en un solo estándar.

```
public class Zoomcameraview extends JavaCameraView {
    public Zoomcameraview(Context context, AttributeSet attrs) { super(context, attrs); }

    protected SeekBar seekBar;

    public void setZoomControl(SeekBar _seekBar) { seekBar = _seekBar; }

    /**
     * Método para activar el zoom de la cámara.
     *
     * @param params Objeto de la clase Camera.Parameters.
     */
    protected void enableZoomControls(Camera.Parameters params) {
        final int maxZoom = params.getMaxZoom();
        seekBar.setMax(maxZoom);
        //Cuando aumentas el zoom se aumenta la visualización de la pantalla.
        seekBar.setOnSeekBarChangeListener(
                new SeekBar.OnSeekBarChangeListener() {
                    int progressvalue = 0;
                    @Override
                    public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
                        progressvalue = progress;
                        Camera.Parameters params = mCamera.getParameters();
                        params.setZoom(progress);
                        mCamera.setParameters(params);
                    }

                    @Override
                    public void onStartTrackingTouch(SeekBar seekBar) {
                    }

                    @Override
                    public void onStopTrackingTouch(SeekBar seekBar) {
                    }
                }
        );
    }
```

Figura 91: Clase Zoomcameraview y el método de aumento de zoom..

Luego creamos un validador si se activó la cámara o no.

```
protected boolean initializeCamera(int width, int height) {
    boolean ret = super.initializeCamera(width, height);
    Camera.Parameters params = mCamera.getParameters();
    if (params.isZoomSupported())
        enableZoomControls(params);
    mCamera.setParameters(params);
    return ret;
}
```

Figura 92: Metodo initializeCamera.

- **Modificación del diseño de la actividad.**

  Para darle el diseño de la pantalla del celular vamos a tener en cuenta 3 cosas:

  - Crear un SeekBar para aumentar el zoom de la cámara.
  - Crear un ImagenButton para tomar las fotos.
  - Visualizador de la cámara a tiempo real con una clase creada.

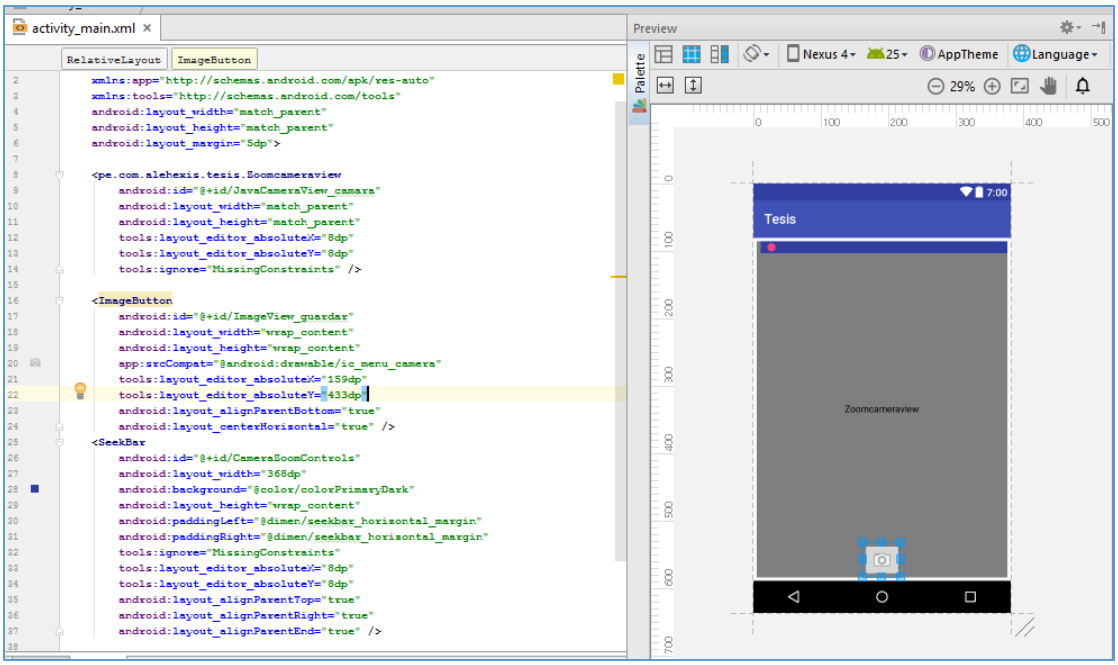

Figura 93: Modelo del diseño de la activity_main.xml

- **Modificación del MainActivity.**

  Para darle la ejecución de cada uno de los eventos del diseño vamos hacer los cambios necesarios de cada componente y la ejecución del método como se almacenará en el dispositivo móvil.

```
/**
 * Clase principal de la ejecucion del app, contiene una implementacion de ejecucion de eventos
 * con el OnClickListener,CameraBridgeViewBase,CvCameraViewListener2
 */
public class MainActivity extends AppCompatActivity implements View.OnClickListener,
        CameraBridgeViewBase.CvCameraViewListener2 {

    private Zoomcameraview JavaCameraViewcamara;
    private Mat mRgba;
    private int alpha = 2, beta = -150;
    private boolean touched = false;
    /**
     * Objeto que verificara el estado del ManagerConnected de la cámara para que haga
     * la visualización de la pantalla en tiempo real.
     * También tiene la opción de la verificación de la importación de la librería de OpenCV
     */
    private BaseLoaderCallback baseLoaderCallback = new BaseLoaderCallback(this) {
        @Override
        public void onManagerConnected(int status) {
            switch (status) {
                case LoaderCallbackInterface.SUCCESS:
                    JavaCameraViewcamara.enableView();
                    break;
                default: {
                    super.onManagerConnected(status);
                }
            }
        }
    };
    private android.widget.ImageButton button;
    private android.widget.ImageView ImageViewguardar;
```

Figura 94: Clase MainActivity.java parte1.

```java
    /**
     * Método para ejecucion de la vista del diseño, aqui se peude hacer
     * las referencias de cada componente de  la vista.
     *
     * @param savedInstanceState objeto de la clase Bundle.
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        this.ImageViewguardar = (ImageView) findViewById(R.id.ImageView_guardar);
        this.JavaCameraViewcamara = (Zoomcameraview) findViewById(R.id.JavaCameraView_camara);
        ImageViewguardar.setOnClickListener(this);
        JavaCameraViewcamara.setVisibility(SurfaceView.VISIBLE);
        JavaCameraViewcamara.setZoomControl((SeekBar) findViewById(R.id.CameraZoomControls));
        JavaCameraViewcamara.setCvCameraViewListener(this);

    }

    /**
     * Método para verificar el estado segundo plano de la actividad.
     */
    @Override
    protected void onPause() {
        super.onPause();
        if (JavaCameraViewcamara != null) {
            JavaCameraViewcamara.disableView();
        }
    }
```

Figura 95: Clase MainActivity.java parte2.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

```java
/**
 * Método para verificar el si la actividad está destruida.
 * Hace que el JavaCameraViewcamara pare al ejecución a tiempo real de la cámara
 * a la visualización,
 */
@Override
protected void onDestroy() {
    super.onDestroy();
    if (JavaCameraViewcamara != null) {
        JavaCameraViewcamara.disableView();
    }
}
/**
 * Cuando la actividad está en segunda plano y se retorna abrir nuevamente.
 * Verifica que si está cargado el módulo de OpenCV para inicializar o ejecutar
 * el onManagerConnected
 */
@Override
protected void onResume() {
    super.onResume();
    if (!OpenCVLoader.initDebug()) {
        OpenCVLoader.initAsync(OpenCVLoader.OPENCV_VERSION_3_2_0, this, baseLoaderCallback);
    } else {
        baseLoaderCallback.onManagerConnected(LoaderCallbackInterface.SUCCESS);
    }
}
/**
 * Método para ejecutar la visualizacion de la camara en un objeto Mat.
 *
 * @param width  - El ancho de los maroos que serán entregados
 * @param height - La altura de los maroos que serán entregados
 */
@Override
public void onCameraViewStarted(int width, int height) {
    //Imgcodecs.CV_LOAD_IMAGE_COLOR --> carga la imagen en el formato BGR.
    mRgba = new Mat(height, width, Imgcodecs.CV_LOAD_IMAGE_COLOR);
}
```

Figura 96: Clase MainActivity.java parte3.

```java
/**
 * Método se utiliza cuando salta el evento de cerrar la cámara.
 */
@Override
public void onCameraViewStopped() {
    mRgba.release();
}

/**
 * Método para cambiar la posición de la cámara en un solo estado.
 *
 * @param imagen             Objeto de la clase Mat.
 * @param mCameraOrientation variable entero que contiene la posición de cambio.
 * @return Objeto de la clase Mat.
 */
private Mat rotarimagen(Mat imagen, int mCameraOrientation) {
    Mat destino = imagen.clone();
    if (mCameraOrientation == 270) {
        Core.flip(imagen.t(), destino, 0);
    } else if (mCameraOrientation == 180) {
        Core.flip(imagen, destino, -1);
    } else if (mCameraOrientation == 90) {
        Core.flip(imagen.t(), destino, 1);
    } else if (mCameraOrientation == 0) {
        destino = imagen;
    }
    return destino;
}
```

Figura 97: Clase MainActivity.java parte4.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

```
    /**
     * Método principal ya que se ejecuta siempre que la cámara este activa y
     * pasa el frame que captura la cámara.
     *
     * @param inputFrame Objeto de la clase CameraBridgeViewBase.CvCameraViewFrame
     * @return Objeto de la clase Mat.
     */
    @Override
    public Mat onCameraFrame(CameraBridgeViewBase.CvCameraViewFrame inputFrame) {
        if (mRgba != null) {
            mRgba.release();}
        mRgba = inputFrame.rgba();
        Mat recorte = mRgba.clone();
        int w = recorte.cols();
        int h = recorte.rows();
        //Marco de referencia para colocar el consumo del medidor.
        Rect ROI = new Rect(h / 2, 10, 100, mRgba.height() / 2 + 120);
        Imgproc.rectangle(recorte, new Point(ROI.x, ROI.y), new Point(ROI.x + ROI.width,
                ROI.y + ROI.height), new Scalar(0, 0, 255), 3);
        if (touched) {
            touched = false;
            Mat result = new Mat(mRgba, ROI);
            guuardar(result);
        }
        return recorte;}

    /**
     * Método para ejecutar los eventos de escuchado de pulsado de un botón o opción.
     * @param v Objeto de la clase View
     */
    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.ImageView_guardar:
                touched = true;
                break;
        }
    }
```

Figura 98: Clase MainActivity.java parte5.

```
/**
 * Método para guardar la imagen segmentada
 *
 * @param imagen_segmentada Objeto de la clase Mat.
 */
private void guuardar(Mat imagen_segmentada) {
    Mat Gris = imagen_segmentada.clone();
    Imgproc.cvtColor(imagen_segmentada, Gris, Imgproc.COLOR_RGB2GRAY);
    Mat RotarImagen = rotarimagen(imagen_segmentada, 90);
    Imgcodecs.imwrite(file("Gray"), RotarImagen);

}

/**
 * Método para nombrar el archivo de la captura.
 *
 * @param nombre Objeto de la clase String que contiene el nombre inicial del método
 * @return Objeto de la clase String  que contiene el nombre final del archivo.
 */
private String file(String nombre) {
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd_HH-mm-ss", Locale.getDefault());
    String currentDateandTime = sdf.format(new Date());
    File mediaStorageDir = new File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES),
            "Tesis");
    if (!mediaStorageDir.exists()) {
        if (!mediaStorageDir.mkdirs()) {
            mediaStorageDir.mkdirs();
        }
    }
    String fileName = mediaStorageDir.getPath() + File.separator + nombre + "_" + currentDateandTime + ".jpg";
    return fileName;
}
```

Figura 99: Clase MainActivity.java parte6.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Basado a la codificación en el método guardar hace la conversión de la imagen original del formato BGR a escala de grises de la imagen segmentada por el marco que codificamos, detallando así el proceso.

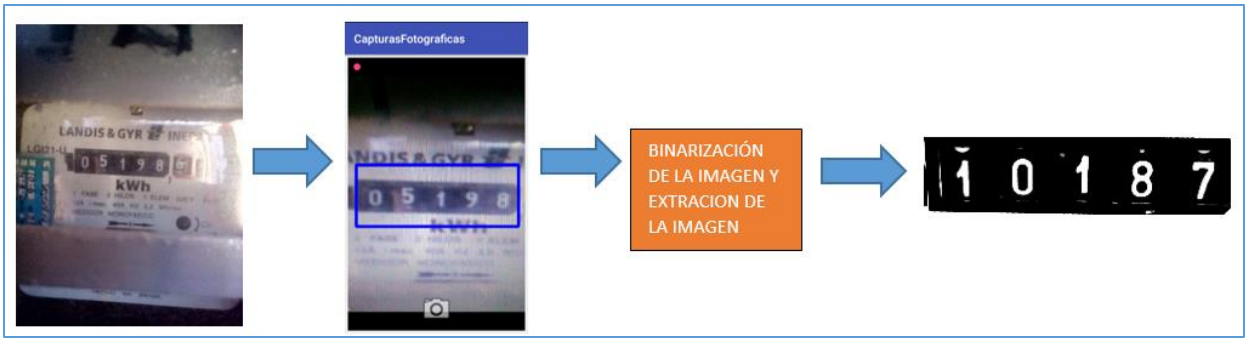

Figura 100: Proceso de toma de fotográfica del aplicación.

7.3. Codificación del aplicativo para el pre procesamiento

Para la codificación del pre procesamiento vamos a utilizar las ubicaciones de la base de datos de la imagen como también la utilización de la librería del OpenCV.

```
static {
    System.loadLibrary(Core.NATIVE_LIBRARY_NAME);
}
static String ubicacionorigen = "D:/Tesis/";
static String dataimagen = ubicacionorigen + "Data/";
static String Resultado_RedNeuronal = ubicacionorigen + "Resultado_RedNeuronal/";
static String bordes = ubicacionorigen + "bordes/";
static String extraer = ubicacionorigen + "Extraer/";
```

Figura 101: Ubicación de la data y su data de salida.

```
private static void Extracion() {
    long TInicio, TFin;
    TInicio = System.currentTimeMillis();
    File file = new File(dataimagen);
    File bordes_Canny = new File(bordes + "Canny");
    File extraer_resultado = new File(extraer + "extraer");
    File bordes_SOBEL = new File(bordes + "SOBEL");
    File bordes_THRESH_BINARY = new File(bordes + "THRESH_BINARY");
    File bordes_THRESH_OTSU = new File(bordes + "THRESH_OTSU");
    if (file.exists()) {
        file.mkdirs();
    }
    EjecutarBorrarCarpeta(bordes_Canny);
    EjecutarBorrarCarpeta(extraer_resultado);
    EjecutarBorrarCarpeta(bordes_SOBEL);
    EjecutarBorrarCarpeta(bordes_THRESH_BINARY);
    EjecutarBorrarCarpeta(bordes_THRESH_OTSU);
    int variable = totalArchivos(file);
```

Figura 102: Localización de las datos de salida.

```
for (int i = 1; i < variable; i++) {
    //Importar la imagen
    Mat image = Imgcodecs.imread(dataimagen + "imagen (" + i + ").jpg", Imgcodecs.CV_LOAD_IMAGE_COLOR);
    if (image.empty() == true) {
        System.out.println("Error no image found!!");
    } else {
        //cambio de tamaño de 250x80
        Imgproc.resize(image, image, new Size(250, 80));
        Mat imageHSV = new Mat(image.size(), CvType.CV_8UC4);
        Mat imageBlurr = new Mat(image.size(), CvType.CV_8UC4);
        //Conversión a escala de gris
        if (image.channels() == 3) {
            Imgproc.cvtColor(image, imageHSV, Imgproc.COLOR_BGR2GRAY);
        }
        Mat imageBlurr1 = imageBlurr.clone();
        Mat ones = Mat.ones(new Size(1, 1), CvType.CV_8U);
        //dilatación de la imagen
        Imgproc.dilate(imageHSV, imageBlurr, ones);
        //erosión de la imagen
        Imgproc.erode(imageBlurr, imageBlurr, ones);
        //Bilateral filtro
        Imgproc.bilateralFilter(imageBlurr, imageBlurr1, 10, 300, 50);
        Mat mat_canny = canny(imageBlurr1);
        //Canny
        Extracciones(mat_canny, image, i, "Canny", 0);
        Mat imageA = new Mat(image.size(), CvType.CV_32F);
        //Binarización
        Imgproc.adaptiveThreshold(imageBlurr1, imageA, 255, Imgproc.ADAPTIVE_THRESH_MEAN_C, Imgproc.THRESH_BINARY, 135, 6);
        Extracciones(imageA, image, i, "THRESH_BINARY", 0);
        //Otsu
        Mat imageA1 = new Mat(image.size(), CvType.CV_32F);
        Imgproc.threshold(imageBlurr1, imageA1, 0, 255, Imgproc.THRESH_OTSU);
        Extracciones(imageA1, image, i, "THRESH_OTSU", 1);
        //sobel
        Mat mat_SOBEL = SOBEL(imageBlurr1);
        Extracciones(mat_SOBEL, image, i, "SOBEL", 0);
```

Figura 103: Proceso del pre procesamiento.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

```
    }
/**
 * Método para eliminar los datos de la carpeta
 * @param fResultadobordes Objeto de la clase File
 */

    private static void EjecutarBorrarCarpeta(File fResultadobordes) {
        String[] hijos = fResultadobordes.list();
        for (String hijo : hijos) {
            new File(fResultadobordes, hijo).delete();
        }
    }
```

Figura 104:Método para eliminar datos de un archivo especifico.

```
    private static Mat canny(Mat imagen) {
        Mat canny = new Mat(imagen.width(), imagen.height(), imagen.type());
        int min_threshold = 100;
        int ratio = 5;
        Imgproc.Canny(imagen, canny, min_threshold, min_threshold * ratio);
        return canny;
    }

    private static Mat SOBEL(Mat imagen) {
        int skala = 1;
        int delta = 0;
        int depth = 3;
        Mat sobel = new Mat(imagen.width(), imagen.height(),imagen.type());
        Mat post_x = new Mat(imagen.width(), imagen.height(), imagen.type());
        Mat post_u = new Mat(imagen.width(), imagen.height(), imagen.type());
        Mat post_x1 = new Mat(imagen.width(), imagen.height(),imagen.type());
        Mat post_u1 = new Mat(imagen.width(), imagen.height(), imagen.type());
        Imgproc.Sobel(imagen, post_x, depth, 1, 0, 3, skala, delta, Core.BORDER_DEFAULT);
        Core.convertScaleAbs(post_x, post_x1);
        Imgproc.Sobel(imagen, post_u, depth, 1, 0, 3, skala, delta,Core.BORDER_DEFAULT);
        Core.convertScaleAbs(post_u, post_u1);
        Core.addWeighted(post_x1, 0.5, post_u1, 0.5, 0, sobel);
        return sobel;
    }

    private static int totalArchivos(File directorio) {
        int total = 0;
        String[] arrArchivos = directorio.list();
        total += arrArchivos.length;
        File tmpFile;
        for (int i = 0; i < arrArchivos.length; ++i) {
            tmpFile = new File(directorio.getPath() + "/" + arrArchivos[i]);
            if (tmpFile.isDirectory()) {
                total += totalArchivos(tmpFile);
            }
        }
```

Figura 105: Utilización del método Canny y Sobel.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

```
private static void Extraciones(Mat mat_canny, Mat iamgen_origen, int posicion, String nombreresultado, int procesa) {
    Mat recorte = iamgen_origen.clone();
    List<MatOfPoint> contours = new ArrayList<>();
    Imgproc.findContours(mat_canny, contours, new Mat(), Imgproc.RETR_CCOMP,
            Imgproc.CHAIN_APPROX_SIMPLE);
    for (int rr = 0; rr < contours.size(); rr++) {
        Rect rect = Imgproc.boundingRect(contours.get(rr));
        if (Imgproc.contourArea(contours.get(rr)) > 100) {
            if (rect.width < rect.height && rect.width > 10
                    && rect.height > 20) {
                Imgproc.rectangle(recorte, new Point(rect.x, rect.y),
                        new Point(rect.x + rect.width, rect.y + rect.height),
                        new Scalar(0, 0, 255), 2);
                if (procesa == 1) {
                    Mat result = iamgen_origen.submat(rect);
                    Imgproc.resize(result, result, new Size(30, 54));
                    Imgproc.cvtColor(result, result, Imgproc.COLOR_BGR2GRAY);
                    Imgproc.blur(result, result, new Size(5, 5));
                    Imgcodecs.imwrite(extraer + "extraer/roi" + posicion + "_"
                            + rr + ".png", result);
                }
            } else {
                Imgproc.rectangle(recorte, new Point(rect.x, rect.y),
                        new Point(rect.x + rect.width, rect.y + rect.height), new Scalar(255, 0, 0), 2);
            }
        }
    }
    Imgcodecs.imwrite(bordes + nombreresultado + "/Resultado_" + nombreresultado + "_" + posicion + ".png", recorte);
}
```

Figura 106: Método para extraer los dígitos reconocidos por los contornos.

Tabla 10: Algoritmos utilizados

Algoritmos utilizados		
bilateralFilter	src	Ubicación entrada de la imagen Objeto Mat.
	dst	Ubicación salida de la imagen Objeto Mat.
	d	diámetro del vecindario considerado
	sigmaColor	en el espacio de color está definido
	sigmaSpace	es el valor sigma en el espacio de coordenadas
dilate	src	Ubicación entrada de la imagen Objeto Mat.
	dst	Ubicación salida de la imagen Objeto Mat.
	kernel	elemento estructurante
erode	src	Ubicación entrada de la imagen Objeto Mat.
	dst	Ubicación salida de la imagen Objeto Mat.
	kernel	elemento estructurante
threshold	src	Ubicación entrada de la imagen Objeto Mat.
	dst	Ubicación salida de la imagen Objeto Mat.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

	thresh	nivel utilizado para el umbral de la imagen
	maxval	se utiliza en los modos Binario y Binary_Inv
	type	constantes de Imgproc utilizadas para describir el tipo de umbral
adaptiveThreshold	src	Ubicación entrada de la imagen Objeto Mat.
	dst	Ubicación salida de la imagen Objeto Mat.
	MAXVALUE	umbral ordinaria
	ADAPTIVEMETHOD	Algoritmo de umbral adaptativo (ADAPTIVE_THRESH_MEAN_C (Calcular la media como la suma del valor de píxel dividida por el número de píxeles) o ADAPTIVE_THRESH_GAUSSIAN_C (ponderación gaussiana para el promedio))
	THRESHOLDTYPE	Tipo de umbral que debe ser THRESH_BINARY o THRESH_BINARY_INV.
	BLOCKSIZE	Tamaño de un vecindario de píxeles que se utiliza para calcular un valor de umbral para el píxel: 3, 5, 7 y así sucesivamente
	C	Constante sustraída de la media o de la media ponderada
findContours	image	Objeto de tipo Mat que es la imagen de entrada
	contours	Lista de la clases MatOfPoint
	hierarchy	Objeto de tipo Mat
	mode	se ocupa de cómo se establecen las relaciones jerárquicas
		RETR_EXTERNAL: da contornos "externos".
		RETR_LIST: no calcula la jerarquía de los contornos.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

		RETR_CCOMP: ordena en modo de jerarquía.
		RETR_TREE: calcula la jerarquía completa de todos los contornos.
	Method	CHAIN_APPROX_NONE: se almacenan todos los puntos de contorno
		CHAIN_APPROX_SIMPLE: Verticales y diagonales se comprimen usando sólo sus puntos finales

Basado a los algoritmos de sus componentes principales ya podíamos especificar un desarrollo hasta la extracción de los números por los contornos siluetados por el Canny, Sobel, Otsu y Binarización.

7.4. Codificación del aplicativo para el entrenamiento.

Se detallará el cómo se procedió el entrenamiento basado a la codificación de esta clase.

```
static {
    System.loadLibrary(Core.NATIVE_LIBRARY_NAME);
}
static int[] valores_prioipales = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'};
static String ubicacionorigen = "D:/Tesis/";
static String extraer = ubicacionorigen + "Extraer/";
static String Resultado_RedNeuronal = ubicacionorigen + "Resultado_RedNeuronal/";

public static void main(String[] args) {
    entrenamiento();
}

private static int totalArchivos(File directorio) {
    int total = 0;
    String[] arrArchivos = directorio.list();
    total += arrArchivos.length;
    File tmpFile;
    for (int i = 0; i < arrArchivos.length; ++i) {
        tmpFile = new File(directorio.getPath() + "/" + arrArchivos[i]);
        if (tmpFile.isDirectory()) {
            total += totalArchivos(tmpFile);
        }
    }
    return total;
}
```

Figura 107: Datos principales de entrenamiento.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

```java
private static void entrenamiento() {
    Scanner keyboard = new Scanner(System.in);
    String ubiicacionarchivo = extraer + "original";
    Ventana im = new Ventana("Entrenamiento");
    File file11 = new File(ubiicacionarchivo);
    //verificar cuantos archivos hay en la carpeta
    int totalArchivos = totalArchivos(file11);
    //valores que solo debe ingresar
    //almacenamiento de los valores
    List<Integer> clasificador = new ArrayList<>();
    //ARCHIVO DONDE SE VA GUADAR
    Mat imagen_train = new Mat();
    //Listar los archivos extraidos
    try {
        for (int posicion = 1; posicion < totalArchivos + 1; posicion++) {
            System.out.println("ejecutando imagen" + posicion);
            Mat image = Imgcodecs.imread(ubiicacionarchivo + "/a (" + posicion + ").png", Imgcodecs.CV_LOAD_IMAGE_COLOR);
            if (image.empty() == true) {
                System.out.println("Error no image found!!");
            } else {
                Mat imageresize = new Mat(image.size(), CvType.CV_8UC4),
                        imageHSV = new Mat(image.size(), CvType.CV_8UC4),
                        morphologyEx = new Mat(image.size(), CvType.CV_8UC4);
                Mat kernel = Imgproc.getStructuringElement(Imgproc.MORPH_ERODE, new Size(3, 3));
                //conversion escala de gris
                if (image.channels() == 3) {
                    Imgproc.cvtColor(image, imageHSV, Imgproc.COLOR_BGR2GRAY);
                }
                //mejoramiento de la imagen con erosion y dilatacion
                Mat dilate = imageHSV.clone();
                Mat ones = Mat.ones(new Size(1, 1), CvType.CV_8U);
                Imgproc.dilate(imageHSV, dilate, ones);
                // morphologyEx mejoramiento de la imagen segun el controno de la imagen para eleminar los contenido de la imagen.
                Imgproc.morphologyEx(dilate, morphologyEx, Imgproc.MORPH_OPEN, kernel);
                //Cambiar el tamaño a 10*10
                Imgproc.resize(morphologyEx, imageresize, new Size(10, 10), 0, 0, Imgproc.INTER_LINEAR);
```

Figura 108: Codificación del método entrenamiento parte1.

```java
//mostrar la imagen
im.showImage(image);
//ingresar el digitos
String input = keyboard.nextLine();
System.out.println(input);
//Validación de la texto ingresado.
if (input.trim().compareTo("exit") != 0) {
    //Verificación si el ingreso es vacio
    if (input.trim().compareTo("") != 0) {
        //conversión de lo ingresado un intchar
        int intChar = (int) input.charAt(0);
        // binarySearch búsqueda a medio intervalo busca el comodín según el arreglo ordenado.
        if (Arrays.binarySearch(valores_prioipales, intChar) >= 0) {
            //ingreso de numero de la clasificacion.
            clasificador.add(intChar);
            //Crear un objeto de tipo Mat
            Mat convertirfloat = new Mat();
            /// CV_32FC1: Define la profundidad de cada elemento y el número de canales
            imageresize.convertTo(convertirfloat, CvType.CV_32FC1);
            //convertir una imagen de tamaño de una fila.
            Mat response_train = convertirfloat.reshape(1, 1);
            //Alamcenamiento de la imagen en otra imagen
            imagen_train.push_back(response_train);
            //almacenamiento
            Imgcodecs.imwrite(Resultado_RedNeuronal + "resize/" + posicion + ".png", imageresize);
            Imgcodecs.imwrite(Resultado_RedNeuronal + "Gris/" + posicion + ".png", imageHSV);
            Imgcodecs.imwrite(Resultado_RedNeuronal + "dilate_erode/" + posicion + ".png", dilate);
            Imgcodecs.imwrite(Resultado_RedNeuronal + "app_cierre/" + posicion + ".png", morphologyEx);
            Imgcodecs.imwrite(Resultado_RedNeuronal + "reshape/" + posicion + ".png", imagen_train);
            continue;
        } else {
            System.out.println("Solo se ingresa numeros del 1 al 9");
        }
    }
} else {
    break;
}
```

Figura 109:Codificación del método entrenamiento parte2.

```
            } else {
                break;
            }
        }
    }
    if (clasificador.size() > 0 && imagen_train.rows() > 0) {
        SacarArchivo(clasificador, imagen_train);
    }
    im.setVisible(false);
} catch (Exception e) {
    e.printStackTrace();
}
```

Figura 110: Codificación del método entrenamiento parte3.

```
private static void SacarArchivo(List<Integer> clasificador, Mat imagen_train) {
    long TInicio, TFin;
    TInicio = System.currentTimeMillis();
    System.out.println("Convertir los datos en xml");
    String dataImages = "";
    //pasar de un integer a string y almacenando en una variable
    for (Integer i : clasificador) {
        dataImages += i + " ";
    }
    String dataClassifications = "";
    //recorrer todo las imagen y almacenar en una variable.
    for (int i = 0; i < imagen_train.rows(); i++) {
        for (int j = 0; j < imagen_train.cols(); j++) {
            double[] temp = imagen_train.get(i, j);
            dataClassifications += temp[0] + " ";
        }
        dataClassifications += "\n";
    }
    //almacenando caracteristicas de la
    String rowsImages = String.valueOf(clasificador.size());
    String colsImages = "1";
    String rowsClassifications = String.valueOf(imagen_train.rows());
    String colsClassifications = String.valueOf(imagen_train.cols());
    DocumentBuilderFactory icFactory_images = DocumentBuilderFactory.newInstance();
    DocumentBuilder icBuilder_images;
```

Figura 111: Codificación del método SacarArchivo.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

```
try {
    icBuilder_images = icFactory_images.newDocumentBuilder();
    Document doc = icBuilder_images.newDocument();
    Element mainRootElement = doc.createElement("opencv_storage");
    doc.appendChild(mainRootElement);
    mainRootElement.appendChild(getMatXML(doc, "classifications", "opencv-matrix", rowsImages, colsImages, "i", dataImages));
    Transformer transformer = TransformerFactory.newInstance().newTransformer();
    transformer.setOutputProperty(OutputKeys.INDENT, "yes");
    DOMSource source = new DOMSource(doc);
    String filename = "src/DataClasificador/classifications.xml";
    File file = new File(filename);
    StreamResult console = new StreamResult(file); //(System.out)
    transformer.transform(source, console);
} catch (Exception e) {
    e.printStackTrace();
}

DocumentBuilderFactory icFactory_classifications = DocumentBuilderFactory.newInstance();
DocumentBuilder icBuilder_classifications;
try {
    System.out.println("entrando a bajar los datos del imagenes");
    icBuilder_classifications = icFactory_classifications.newDocumentBuilder();
    Document doc = icBuilder_classifications.newDocument();
    Element mainRootElement = doc.createElement("opencv_storage");
    doc.appendChild(mainRootElement);
    mainRootElement.appendChild(getMatXML(doc, "images", "opencv-matrix", rowsClassifications,
            colsClassifications, "f", dataClassifications));
    Transformer transformer = TransformerFactory.newInstance().newTransformer();
    transformer.setOutputProperty(OutputKeys.INDENT, "yes");
    DOMSource source = new DOMSource(doc);
    String filename = "src/DataClasificador/images.xml";
    File file = new File(filename);
    StreamResult console = new StreamResult(file); //(System.out)
    transformer.transform(source, console);
} catch (Exception e) {
    e.printStackTrace();
}
```

Figura 112: Codificación del método SacarArchivo.

```
private static Node getMatXML(Document doc, String option_id, String type_id, String rows, String cols, String dt, String data) {
    Element elem = doc.createElement(option_id);
    elem.setAttribute("type_id", type_id);
    elem.appendChild(getMatXMLElement(doc, "rows", rows));
    elem.appendChild(getMatXMLElement(doc, "cols", cols));
    elem.appendChild(getMatXMLElement(doc, "dt", dt));
    elem.appendChild(getMatXMLElement(doc, "data", data));
    return elem;
}

private static Node getMatXMLElement(Document doc, String name, String value) {
    Element node = doc.createElement(name);
    node.appendChild(doc.createTextNode(value));
    return node;
}
```

Figura 113: Métodos para genera la estructura de la data.

7.5. Codificación del aplicativo para ejecutar los resultados

Para ejecutar los resultados se ha tenido que crear un interfaz donde selecciona los datos del entrenamiento y luego la imagen para dar conformidad del resultado

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

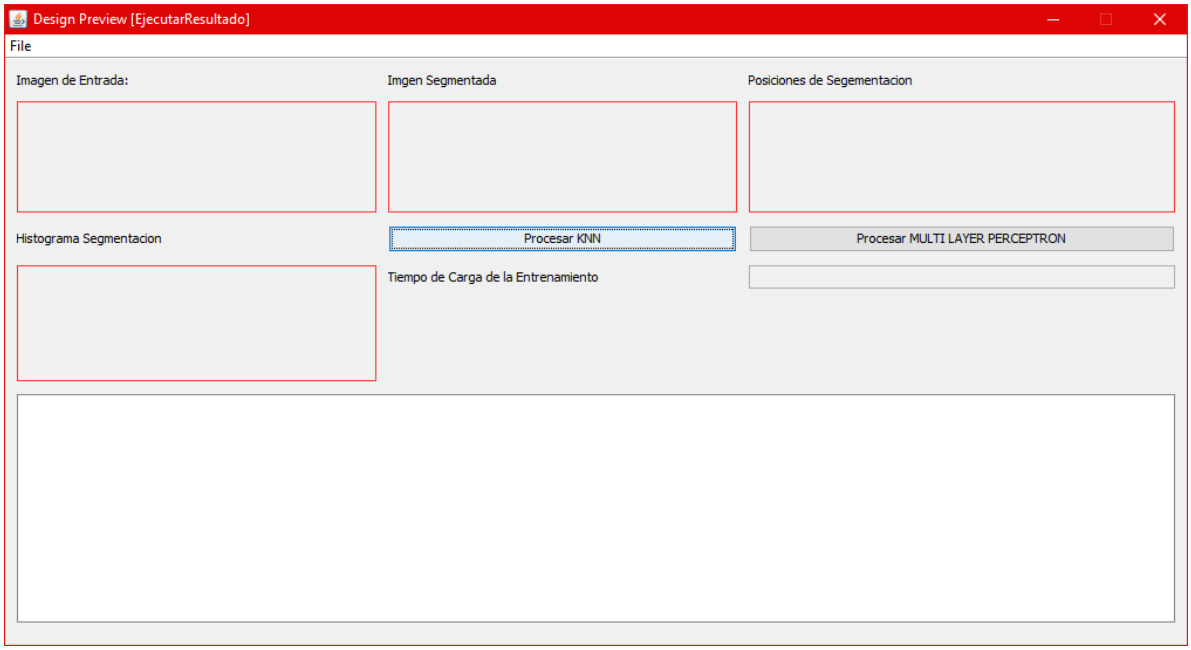

Figura 114: Diseño de la interfaz para la ejecución de los resultados.

```
private Mat Train, image;
private List<Integer> label_data;
private final JFileChooser fileChooser;
private final KNearest kNearest;
private final int[] valores_pricipales = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'};

public EjecutarResultado() {
    System.loadLibrary(Core.NATIVE_LIBRARY_NAME);
    initComponents();
    fileChooser = new JFileChooser();
    kNearest = KNearest.create();
}

public static BufferedImage createAwtImage(Mat mat) {

    int type = 0;
    switch (mat.channels()) {
        case 1:
            type = BufferedImage.TYPE_BYTE_GRAY;
            break;
        case 3:
            type = BufferedImage.TYPE_3BYTE_BGR;
            break;

    }
    BufferedImage image = new BufferedImage(mat.width(), mat.height(), type);
    WritableRaster raster = image.getRaster();
    DataBufferByte dataBuffer = (DataBufferByte) raster.getDataBuffer();
    byte[] data = dataBuffer.getData();
    mat.get(0, 0, data);

    return image;
}
```

Figura 115: Datos principales del ejecución.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

```
private List ExtraerListLabel(String srcClasificadorclassificationsxml) {
    List<Integer> data_label = null;
    try {
        File fXmlFile = new File(srcClasificadorclassificationsxml);
        if (fXmlFile.exists()) {
            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
            Document doc = dBuilder.parse(fXmlFile);
            doc.getDocumentElement().normalize();
            NodeList nList = doc.getElementsByTagName("classifications");
            for (int temp = 0; temp < nList.getLength(); temp++) {
                Node nNode = nList.item(temp);
                if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                    Element eElement = (Element) nNode;
                    int row = Integer.parseInt(eElement.getElementsByTagName("rows").item(0).getTextContent());
                    int cols = Integer.parseInt(eElement.getElementsByTagName("cols").item(0).getTextContent());
                    String[] data = eElement.getElementsByTagName("data").item(0).getTextContent().split(" ");
                    data_label = new ArrayList();
                    for (int i = 0; i < data.length; i++) {
                        data_label.add(Integer.parseInt(data[i]));

                    }
                }
            }
        } else {
            data_label = null;
        }
    } catch (ParserConfigurationException | SAXException | IOException ex) {
        ex.printStackTrace();
    }
    return data_label;
}
```

Figura 116: Método de convertir un variable a Mat.

```
private synchronized static double[][] ExtraerTrain(String srcClasificadorclassificationsxml) {
    double[][] data_float = null;
    try {
        File fXmlFile = new File(srcClasificadorclassificationsxml);
        if (fXmlFile.exists()) {
            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
            Document doc = dBuilder.parse(fXmlFile);
            doc.getDocumentElement().normalize();

            NodeList nList = doc.getElementsByTagName("images");
            for (int temp = 0; temp < nList.getLength(); temp++) {
                Node nNode = nList.item(temp);
                if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                    Element eElement = (Element) nNode;
                    int row = Integer.parseInt(eElement.getElementsByTagName("rows").item(0).getTextContent());
                    int cols = Integer.parseInt(eElement.getElementsByTagName("cols").item(0).getTextContent());
                    String[] data = eElement.getElementsByTagName("data").item(0).getTextContent().split("\n");
                    data_float = new double[row][cols];
                    for (int i = 0; i < data.length; i++) {
                        String[] data1 = data[i].split(" ");
                        for (int j = 0; j < data1.length; j++) {
                            data_float[i][j] = Double.parseDouble(data1[j]);
                        }
                    }
                    Mat Train = Mat.zeros(row, cols, CvType.CV_32F);
                    for (int i = 0; i < data_float.length; i++) {
                        for (int j = 0; j < data_float[i].length; j++) {
                            Train.put(i, j, data_float[i][j]);
                        }
                    }
                }
            }
        } else {
            data_float = null;
        }
```

Figura 117: Método de convertir un variable a Mat.

```
private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
    long inicio = System.currentTimeMillis();
    label_data = ExtraerListLabel("src/DataClasificador/classifications.xml");
    double[][] ExtraerTrain = ExtraerTrain("src/DataClasificador/images.xml");
    Train = Mat.zeros(ExtraerTrain.length, 100, CvType.CV_32F);
    for (int i = 0; i < ExtraerTrain.length; i++) {
        for (int j = 0; j < ExtraerTrain[i].length; j++) {
            Train.put(i, j, ExtraerTrain[i][j]);
        }
    }
    if (Train.dump().length() > 0) {
        JOptionPane.showMessageDialog(this, "Datos Cargados");
        kNearest.train(Train, Ml.ROW_SAMPLE, Converters.vector_int_to_Mat(label_data));
        jTextField_CragaDatos.setText(((System.currentTimeMillis() - inicio) / 1000) + " segundos");
    }
}

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
    if (fileChooser.showDialog(null, "Abrir imagen") == JFileChooser.APPROVE_OPTION) {
        jTextArea_Resultado.setText("");
        jTextField1.setText("");
        Label_ImagenResultado.setText("");
        Label_ImagenResultado1.setText("");
        File archivoElegido = fileChooser.getSelectedFile();
        String parent = archivoElegido.getPath();
        image = Imgcodecs.imread(parent, Imgcodecs.CV_LOAD_IMAGE_COLOR);
        ImageIcon image1 = new ImageIcon(createAwtImage(image));
        Label_ImagenEntrada.setIcon(image1);
    }
}
```

Figura 118: Método para cargar la data y la imagen.

Para ejecutar los resultados se utilizó la parte del pre procesamiento de las imágenes para comprobar con el KNN los resultados.

```
    List<MatOfPoint> contours = new ArrayList<>();
    Imgproc.findContours(imageBlur1, contours, new Mat(), Imgproc.RETR_CCOMP, Imgproc.CHAIN_APPROX_SIMPLE);
    for (int rr = 0; rr < contours.size(); rr++) {
        Rect rect = Imgproc.boundingRect(contours.get(rr));
        if (Imgproc.contourArea(contours.get(rr)) > 100) {
            if (rect.width < rect.height) {
                Imgproc.rectangle(recorte, new Point(rect.x, rect.y), new Point(rect.x + rect.width, rect.y
                        + rect.height), new Scalar(0, 0, 255), 2);
                Mat result = image.submat(rect);
                Imgproc.resize(result, result, new Size(10, 10));
                if (result.channels() == 3) {
                    Imgproc.cvtColor(result, result, Imgproc.COLOR_BGR2GRAY);
                }
                Mat convertirfloat = new Mat();
                result.convertTo(convertirfloat, CvType.CV_32F);
                Mat response_train = convertirfloat.reshape(1, 1);
                float p = kNearest.findNearest(response_train, 2, new Mat());
                int binarySearch = Arrays.binarySearch(valores_pricipales, (int) p);
                resultado += "Numero: " + binarySearch + " -Probalidad:" + (kNearest.predict(response_train)) + "\n";
                arreglo += binarySearch + "-";
                Imgproc.putText(recorte1, binarySearch + "",
                        new Point(rect.x, rect.y + rect.height), Core.FONT_HERSHEY_SIMPLEX, 1.0, new Scalar(255, 0, 0));
            }
        }
    }

    jTextArea_Resultado.setText(resultado + "\n" + ((System.currentTimeMillis() - inicio) / 1000) + " segundos");
    jTextField1.setText(arreglo.substring(0, arreglo.length() - 1));
    ImageIcon image1 = new ImageIcon(createAwtImage(recorte));
    Label_ImagenResultado.setIcon(image1);
    ImageIcon image = new ImageIcon(createAwtImage(recorte1));
    Label_ImagenResultado1.setIcon(image);
}
```

Figura 119: Codificación de los resultado de la imagen.

Basado a eso al momento de ejecutar saldrá así los resultados.

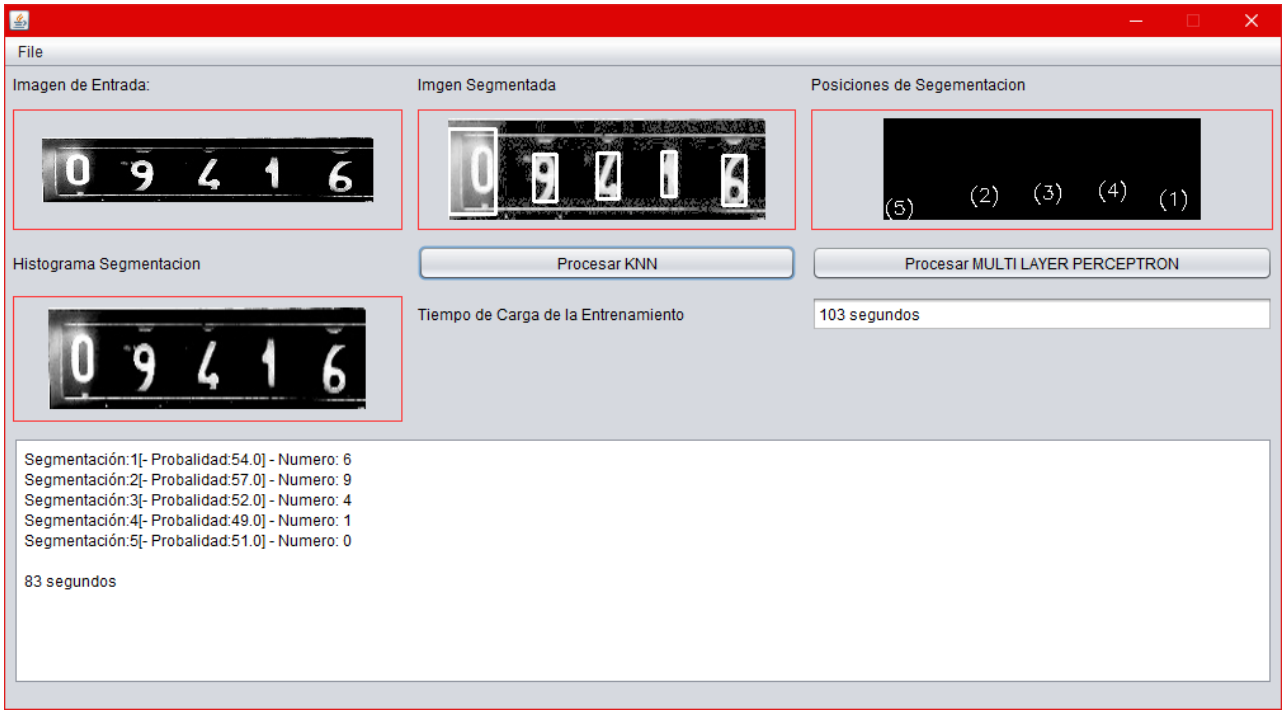

Figura 120: Resultado al ejecutar el entrenamiento.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

BIBLIOGRAFÍA

Abdelmalik, M., Inza, I., & Larrañaga, P. (2015). Tema 5. Clasificadores K-NN, 1–8. Retrieved from http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/t9knn.pdf

Aguilera, S. O., & Constenla, G. (2010). Aplicación de Redes Neurales en el reconocimiento óptico de caracteres. *Grupo de Software Libre – Facultad de Tecnología Informática – Universidad de Belgrano*, 100–104.

Aparicio Martín de Loeches, A., & Fernández Guzmán, L. (2014). Reconocimiento óptico de caracteres en imágenes digitales de contadores de gas. Retrieved from http://eprints.ucm.es/31485/1/Memoria Final.pdf

Bejarano Hidalgo, I., & Sánchez García de Blas, R. J. (2015). *Reconocimiento de caracteres mediante imágenes en contadores de gas en entornos reales*. Universidad Complutense de Madrid.

Campos Aquino, D. A., & Mundaca Arriola, L. A. N. (2016). *Propuesta de método de reconocimiento de imágenes para la identificación del melanoma humano*. Universidad Señor de Sipán. Retrieved from http://repositorio.uss.edu.pe/bitstream/uss/156/1/4TESIS.pdf

Caponetti, L., & Castellano, G. (2017). *Fuzzy Logic for Image Processing* (Springer). https://doi.org/10.1007/978-3-319-44130-6

Carranza Hernández, S. N. (2014). *Implementación de un sistema de información para el reconocimiento de caracteres basado en la red neuronal perceptron*. PUCP. Retrieved from http://tesis.pucp.edu.pe/repositorio/handle/123456789/5956

Casillas Gil, N. (2012). *Sistema Basado de redes neuronales para el reconocimiento de dígitos manuscritos*. UNIVERSIDAD CARLOS III DE MADRID. Retrieved from http://e-archivo.uc3m.es/handle/10016/15011

Chancafe Sirlopu, J. R., & Mazabel Quijandria, G. A. (2016). *Detección automática de caries utilizando reconocimiento de patrones en placas*

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

radiográficas. Universidad Señor de Sipán. Retrieved from http://repositorio.uss.edu.pe/handle/uss/163

Flores Quispe, R. (2006). *Reconocimiento De Caracteres Manuscritos Aislados En Campos De Formularios Utilizando Redes Neuronales*. UNIVERSIDAD NACIONAL DE INGENIERIA.

Garbi, J. L., Mercado, P., Lanzarini, L., & Russo, C. (2007). Reconocimiento de Números Manuscritos. *Congreso Argentino de Ciencias de La Computación*, 733–742.

Garrido Rojas, E. R. (2010). *Diseño e implementación de un módulo de reconocimiento de número manuscritos*. PUCP. Retrieved from http://tesis.pucp.edu.pe/repositorio/handle/123456789/903

Gómez González, S., & Gutierrez Alzate, S. (2011). *Desarrollo de un sistema prototipo de reconocimiento de dígitos usando momentos invariantes*. UNIVERSIDAD TECNOLÓGICA DE PEREIRA.

Islam, N., Islam, Z., & Noor, N. (2017). A Survey on Optical Character Recognition System, *10*(December), 1–4. Retrieved from http://arxiv.org/abs/1710.05703

Lélis, D. (2015). *OpenCV 3 . 0 Computer Vision with Java* (I). Birmingham: Packt Publishing Ltd. Retrieved from www.packtpub.com

Lenovo. (2016). Detalle de celular Lenovo A2010. Retrieved July 9, 2017, from http://shop.lenovo.com/co/es/smartphones/serie-a/a2010/

Lucero Carrillo, E. D., & Saldaña Saldaña, H. A. (2016). *Utilización de técnicas de visión artificial para la detección automática de defectos externos del mango*. Universidad Señor de Sipán. Retrieved from http://repositorio.uss.edu.pe/handle/uss/162

OpenCV. (2017). Opencv Documentación. Retrieved July 17, 2017, from http://docs.opencv.org/2.4/index.html

Romero, D. J., Seijas, L., & Ruedin, A. (2006). *Reconocimiento de Dígitos*

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Manuscritos Usando La Transformada Wavelet Continua en 2 Dimensiones y Redes Neuronales. *Paper de Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires*. Retrieved from http://sedici.unlp.edu.ar/bitstream/handle/10915/22473/Documento_completo.PDF?sequence=1

Salas Damián, R. C. (2007). *Diseño de un corrector ortográfico para un sistema de reconocimiento opticos de caracteres*. PUCP.

Sánchez Fernández, C. J., & Sadonís Consuegra, V. (2009). Reconocimiento Óptico de Caracteres (OCR), 7. Retrieved from http://www.it.uc3m.es/jvillena/irc/practicas/08-09/09.pdf

Tito Chura, E., & Silva Delgado, C. (2015). The species of the euphorbiaceae family in tacna province: biosystematic study. *Ciencia de Desarrollo*, *I*, 61–67. Retrieved from revistas.unjbg.edu.pe/index.php/CYD/article/download/410/361

Vassallo Barco, M. J. (2015). *PROCESAMIENTO DE IMÁGENES DIGITALES UTILIZANDO DESCRIPTORES DE FORMA PARA LA IDENTIFICACIÓN DE DEFICIENCIAS NUTRICIONALES A NIVEL FOLIAR DEL CAFETO*. Universidad Señor de Sipán.

Vilcherrez Chavarry, K. L. (2015). *PROCESAMIENTO DE IMÁGENES DIGITALES UTILIZANDO DESCRIPTORES DE COLOR PARA LA IDENTIFICACIÓN DE DEFICIENCIAS NUTRICIONALES EN EL CAFÉ A NIVEL FOLIAR*. Universidad Señor de Sipán.

Wainschenker, R., Massa, J. M., & Tristan, P. (2011). *Procesamiento Digital de Imágenes*. *Optativa Área Procesamiento de señales Primer cuatrimestre de* (Vol. 43). https://doi.org/10.1016/j.scriptamat.2007.01.031

Xataka. (2016). Huawei P8 Lite Smart. Retrieved July 9, 2017, from https://www.xatakamovil.com/otras/huawei-p8-lite-smart-nuevo-procesador-para-la-renovacion-del-superventas-de-huawei

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Yamaguchi, T., Nakano, Y., Maruyama, M., Miyao, H., & Hananoi, T. (2003). Digit Classification on Signboards for Telephone Number Recognition. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, *2003–Janua*(Icdar), 359–363. https://doi.org/10.1109/ICDAR.2003.1227689

LISTA DE FIGURAS

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

LISTA DE TABLAS

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe