

















































































































En la figura 4.11 se muestra el diagrama de flujos completo de toda la etapa de visión de este estudio, desde la lectura de la imagen o captura hasta la señal de orden al SME.

### **4.3.2 Captura de la Imagen**

Para la obtención de la imagen se utilizará la cámara seleccionada en el apartado anterior. El software Matlab donde se desarrollará el algoritmo de visión debe contar con el driver PointGrey perteneciente al fabricante de la cámara, mediante el cual es posible enlazar la cámara y el software. Logrando así la captura en línea de las imágenes.

Para capturar la imagen en el momento adecuado, el cual es cuando la cámara captura totalmente el racimo de uva, se utilizó un disparo por software de la cámara. Este disparo por software, comienza con el procesamiento de la faja sin detección de uva, mediante filtros y binarización, generando un valor de 0 cada vez que no se detecte racimo de uva, al detectar la mínima entrada de un racimo, generara un valor de 1 para realizar la captura 1.2 segundos después de haber generado este valor de 1. El valor de 1.2 segundos se realizó mediante pruebas experimentales en el sistema de pruebas, siendo el valor óptimo para la captura de la imagen.

### **4.3.3 Preprocesamiento y Segmentación de la Imagen**

En este apartado, se realizará la metodología para el desarrollo de las etapas de preprocesamiento y segmentación de la imagen, en el cual, la entrada en esta etapa es la imagen monocromática capturada por la cámara y como salida serán el área, tamaño en pixeles del racimo de uva. En la figura 4.15, se muestra esta etapa como una caja negra, esta caja negra se dividió en funciones que se mostró en el diagrama de flujo mostrado en la figura 4.13.

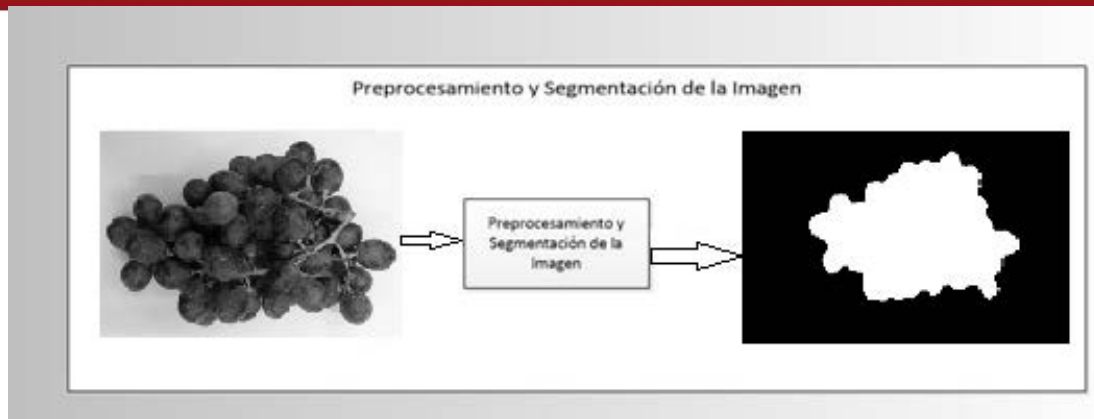


Figura 4.15 Caja Negra Preprocesamiento Sistema de Visión

A continuación, se detallará cada etapa dentro de la caja negra.

#### 4.3.3.1 Filtrado

En esta etapa se aplicará un filtro espacial a la imagen capturada para mejorar la imagen y eliminar los posibles problemas de ruido que contenga la captura. A continuación se realiza el proceso de esta etapa mediante imágenes.



Figura 4.16 Imagen original capturada (I1)

Al capturar la imagen se aprovechará la gran diferencia en gris, entre el racimo de uva y la faja, a continuación se muestra el histograma de la figura 4.16

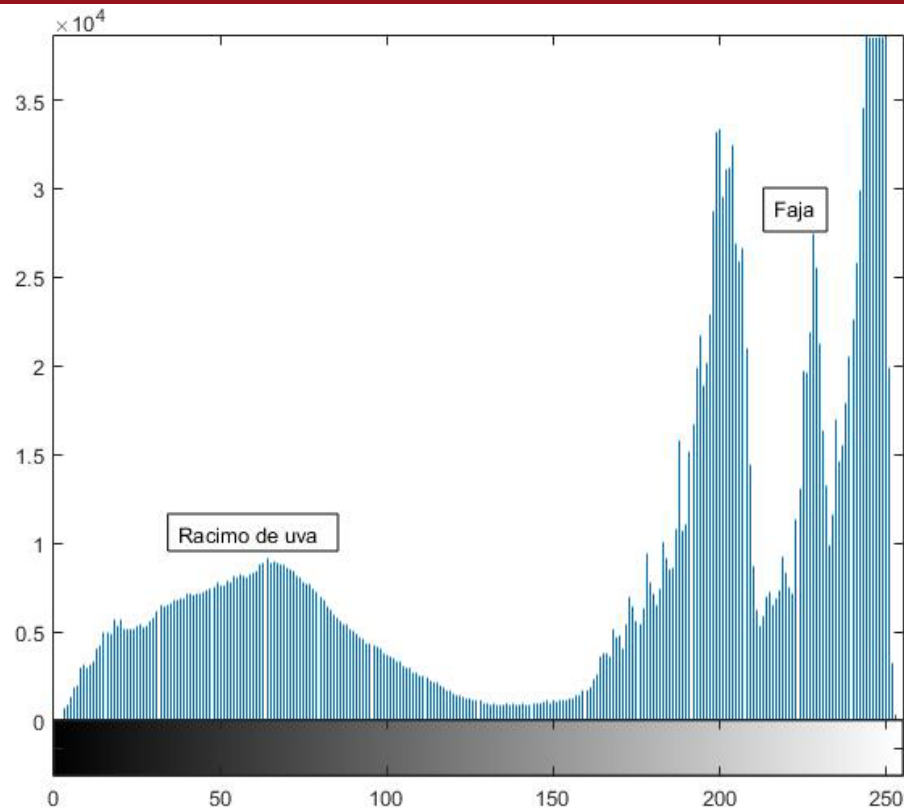


Figura 4.17 Histograma de la imagen I1

En la figura 4.17 se muestra las diferencias que existen en el histograma de la imagen original, donde el eje horizontal que va de 0 a 255 es el nivel de gris que presenta la imagen, siendo 0 el color negro y 255 el color blanco. El eje vertical indica la cantidad de píxeles a un determinado nivel de gris que existe en la imagen. Como se puede observar esta imagen tiene altos valores de blanco debido a que la mayoría de píxeles en la foto pertenece a la faja que es de un color claro a diferencia del racimo de uva.

$$\mathbf{Filtro\ espacial} = \frac{1}{4} \cdot \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Este filtro espacial se aplicó en la imagen I1 generando la figura 4.18. Este filtro se aplica debido al valle que existe en el histograma, lo que va generar una mejor separación entre el racimo de uva y la faja, como se muestra en la figura 4.18.



Figura 4.18 Racimo de uva separación entre racimo y faja (I2)

Se observa en la figura 4.18, una buena separación entre la faja y el racimo de uva, pero hay puntos dentro de las bayas que tienen la misma cantidad de brillo que la faja lo que origina huecos blancos dentro del racimo, lo que no es adecuado para este estudio. Por lo tanto para eliminar esos huecos dentro del racimo se utilizaron operaciones morfológicas como apertura, para obtener un racimo completo que no contenga menos huecos.

La siguiente operación de filtrado que se realiza es un filtro llamado apertura, la cual fue tratada en el capítulo anterior, a continuación se muestra la imagen del racimo I2 luego de aplicar el filtro mencionado. Se observa la disminución en la cantidad de huecos del racimo, posteriormente se aplicara otro filtro en la etapa de binarización.



Figura 4.19 Racimo de uva luego de aplicar filtro apertura cierre (I3)

Según la figura 4.19, se observa que los brillos se oscurecen en los bordes para poder realizar una buena segmentación. En la imagen I2, existen aún pequeños bordes diferentes de valor de 255 en la imagen lo cual ocasionará una incorrecta segmentación. Por ello al usar este filtro se oscurecerá los pequeños bordes para luego realizar una umbralización correcta.

#### 4.3.3.2 Umbralización de la Imagen

Luego de obtener la imagen I3, se debe realizar la etapa de umbralización para poder obtener una imagen I4 en formato binario (1,0), donde el racimo se identificará por número 1 y el fondo por número 0, para este propósito se seleccionó que para un umbral de 255 se denote como fondo y valores menores a 255 para el racimo de uva.

$$I4 = I3 < 255;$$

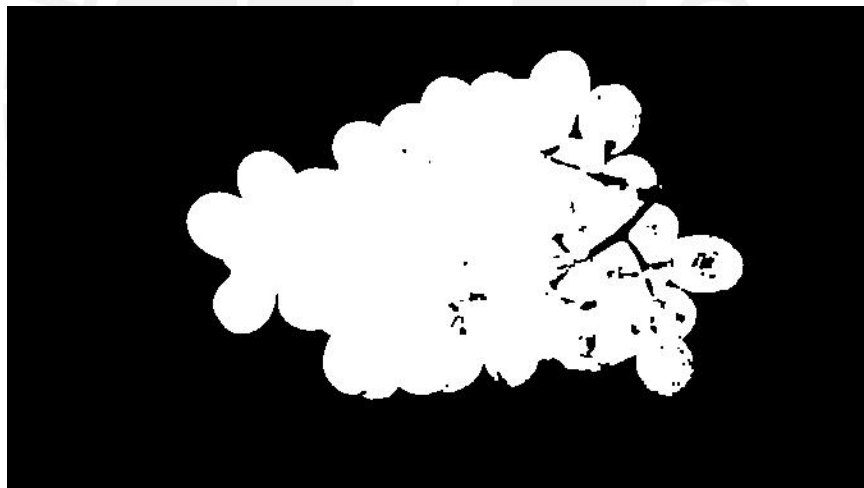


Figura 4.20 Umbralización de imagen I3

En la figura 4.20, se observa una posible segmentación del racimo. Esta segmentación presenta problemas ya que hay huecos en el racimo, así como las ramas del racimo que no se pudieron segmentar.

Para poder eliminar el problema presentado luego de la segmentación se procederá a hacer uso de operaciones morfológicas para poder eliminar los huecos negros en la imagen binaria, y poder capturar de una mejor forma todo el tamaño completo del racimo. Se procede a realizar las operaciones morfológicas sobre la imagen I4.

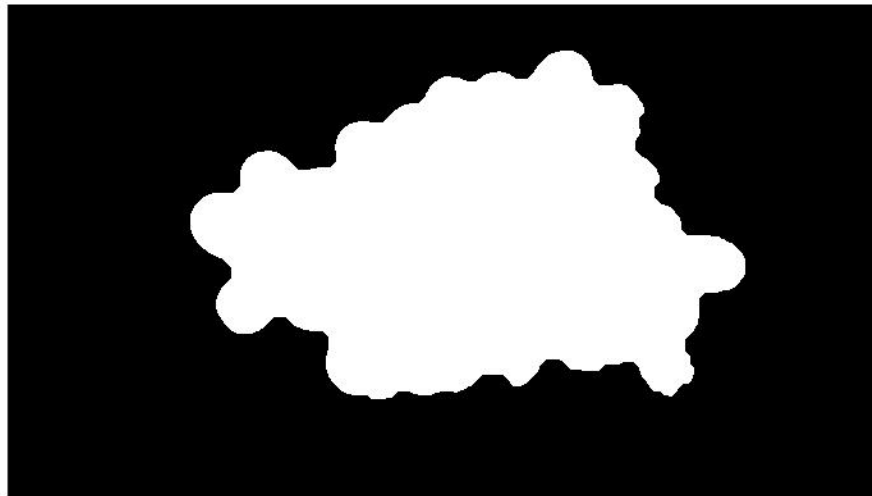


Figura 4.21 Operaciones morfológicas sobre I4 (I5)

Según la figura 4.21, se puede observar que se eliminaron correctamente los huecos negros, que ocasionaban una mala segmentación. Ahora es posible extraer características del racimo como puede ser el tamaño y el área que ocupa el racimo dentro de la imagen capturada. En el siguiente apartado se tratará de extraer los descriptores mencionados, para luego poder determinar mediante redes neuronales el peso del racimo.

#### 4.3.4 Cálculo de Descriptores

En esta etapa se procede a extraer información importante para introducir al sistema de reconocimiento. Los descriptores o la información más resaltante para poder cumplir con los objetivos de este trabajo de tesis son, el cálculo del Área y el Bounding Box.

A continuación se describirá cada etapa en este estudio y como se abordó su solución.

#### 4.3.4.1 Cálculo de Área

El cálculo del área de la imagen I5, sería el número de píxeles que contiene el racimo. Esta información debe ser extraída ya que es un requisito de entrada para la red neuronal.

Contando los píxeles con valor de 1 en la imagen I5 se obtiene un valor de 674839 píxeles que representaría el valor del área.

#### 4.3.4.2 Bounding Box

Uno de los factores importantes y que es requerido en el sistema de visión es el tamaño del racimo de uva, como el ancho y largo. Por ello hacer uso del bounding box (bbox) que circunscribe al racimo, sería la forma más rápida y simple de solucionar lo requerido. Como se explicó en el capítulo anterior el bbox es el mínimo rectángulo que encierra al racimo de uva, por ello sabiendo el tamaño del bbox sería suficiente para conocer el tamaño del racimo.

Luego de realizar este procedimiento se consigue el ancho y largo del racimo, todos estos valores fueron calculados en píxeles, pero las medidas reales no deben estar en píxeles sino en sistema internacional centímetros o metros.

#### 4.3.4.3 Conversión de píxeles a sistema internacional

En este apartado se convertirá las medidas halladas en el subcapítulo anterior en valores reales como lo es el sistema internacional, para ello en una de las imágenes se capturó la imagen de una regla para poder hacer la transformación de píxeles a centímetros.

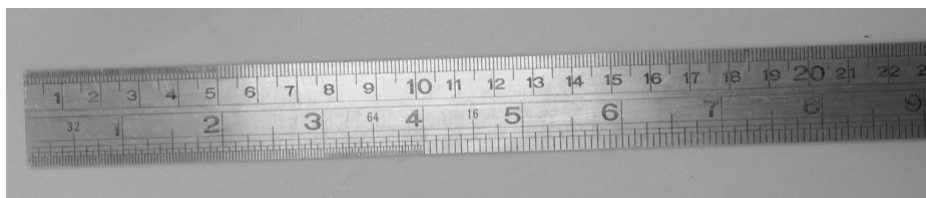


Figura 4.22 Imagen Patrón regla

En la figura 4.22, se muestra la imagen de una regla capturada a las mismas condiciones que los racimos capturados, Esta imagen se tomó en el centro del área cubierta por la cámara. esto se hizo con el propósito de poder determinar la cantidad de pixeles que hay en una longitud determinada.

Luego de analizar la figura 4.22, se comprueba que cada 10 centímetros en la regla patrón significan 540 pixeles, por lo cual determinar el tamaño real del racimo o del rectángulo que encierra el racimo de uva será factible.

#### **4.3.5 Diseño Perceptrón Multicapa**

En esta parte, el propósito principal consiste en diseñar y construir una Red Neuronal MLP, que permita clasificar los racimos de uva según los descriptores extraídos en el capítulo anterior. Estos descriptores se evaluarán en línea con la red entrenada, generando así una respuesta de peso, permitiendo luego la clasificación de los racimos según estándares de calidad por peso.

En el capítulo 3, se conceptualizó sobre el tipo de red neuronal a utilizar en este estudio, desde su principio básico llamado Perceptrón simple a MLP. Adicionalmente se planteó sobre los pasos para implementar la red neuronal MLP y el algoritmo BP, que se utilizara en este trabajo.

A continuación, se detallará la forma de cómo se abordó el diseño de esta red neuronal.

##### **4.3.5.1 Generación de Base de datos y Escalamiento**

En esta etapa, se requiere ordenar los datos de entrada para la red neuronal, ordenando las características en un vector así como los pesos en un vector de salidas, Para la generación de la base de datos se utilizaron una cantidad de imágenes para el entrenamiento de data y se reservó otra cantidad para la validación de la red neuronal.



Figura 4.23 Racimos de uva para entrenamiento

Para la obtención de los pesos de las uvas de mesa se utilizó una balanza electrónica. El peso es requerido ya que la red neuronal utilizada es una red supervisada, que necesita conocer entradas y salidas para el entrenamiento.

A continuación se mostrará el acomodamiento de los descriptores como un vector de entrada y los pesos como un vector de salida para una muestra dada.

$$x = \begin{pmatrix} \text{Largo} \\ \text{Ancho} \\ \text{Area} \end{pmatrix} = \begin{pmatrix} 1303 \\ 814 \\ 674839 \end{pmatrix}$$

$$y = [\text{Peso}] = [448]$$

La información de los descriptores así como los pesos se deben normalizar, a valores comprendidos entre 0 y 1, esto se debe a que se usarán funciones de activación que varían en ese rango como lo son la función tansig y logsig. Por ello hacer un escalamiento de la data es requerido.

Para escalar el ancho y largo se utilizarán un factor de multiplicación de  $10^{-3}$  y para el área se usarán factores de  $10^{-6}$ . Para escalar el peso se utilizarán un factor de  $10^{-3}$ . Por lo tanto la data utilizada para entrenamiento, cambiará a la siguiente forma mostrada.

$$x = \begin{pmatrix} \text{Largo} \\ \text{Ancho} \\ \text{Area} \end{pmatrix} = \begin{pmatrix} 0.1303 \\ 0.814 \\ 0.674839 \end{pmatrix}$$

$$t = [Peso] = [0.448]$$

Por lo tanto para toda la data que se cuenta se tendrá una data en forma vectorial de la siguiente forma:

$$[X, T] = \{(x_1, t_1), (x_2, t_2), \dots (x_n, t_n)\}$$

Donde  $[X, T]$ , es la data total de entrenamiento que ingresara a la RNA, donde X es la data de entrada a la red neuronal y T es el objetivo que se quiere alcanzar luego de entrenada la red que en este estudio es el peso del racimo.

#### 4.3.5.2 Arquitectura red neuronal

En esta etapa luego de normalizar los valores de entrada y salida de la red supervisada, se escogerá la arquitectura de la red neuronal. La arquitectura que se seleccionara dependerá de la velocidad de entrenamiento y el error de generalización, tomando en cuenta diferente cantidad de neuronas en la capa oculta.

El modelo del MLP estará constituido por 4 capas, una capa de entrada, dos capas ocultas y una capa de salida. La capa de entrada consistirá de 3 neuronas que es igual al tamaño del vector de descriptores, que proviene de la imagen. Las capas ocultas consisten de 5 neuronas en cada capa, este valor puede ser modificado.

La capa de salida, es igual al número de salidas que cuenta el sistema, para este estudio es una sola salida por lo tanto, la capa de salida consistirá de una solo neurona.

A continuación se muestra la arquitectura de la RNA que se utilizará en este estudio.

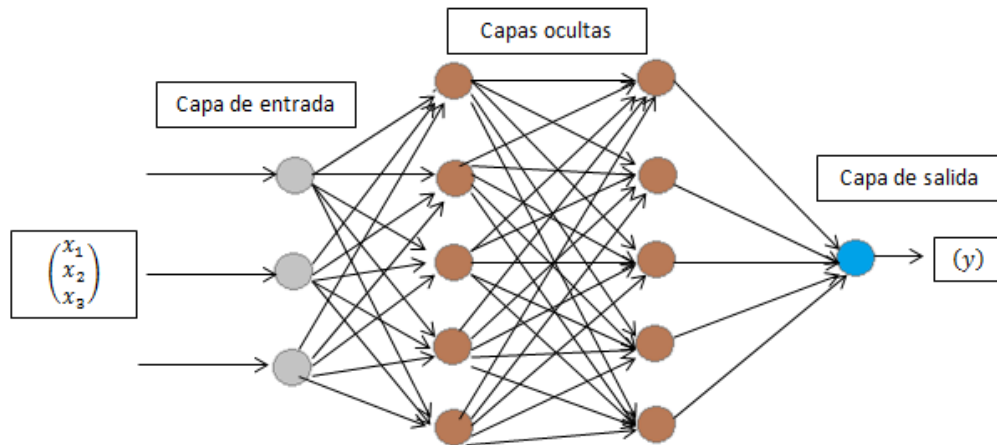


Figura 4.24 Estructura de red MLP utilizada

#### 4.3.5.3 Etapa de entrenamiento

En este apartado se describirá, la etapa de entrenamiento de la arquitectura seleccionada en el apartado anterior. Para el entrenamiento se hizo uso del entorno matlab para conseguir la red entrenada, luego los valores de pesos y bias de las neuronas en cada capa son extraídas para la etapa de validación. Este entrenamiento logra converger cuando se alcanza una cierta cantidad de épocas o se logra un error mínimo que es introducido en el código matlab. Las funciones de activación que se usaron para esta arquitectura fueron tansig y logsig. A continuación se muestra la interfaz que se utilizó para el entrenamiento y un extracto del código matlab.


 Figura 4.25 Entrenador de red neuronal diseñado  
 Fuente propia

```

P=double(X);

T;
PR=minmax(P);
net = newff(PR,[5 5 1],{'tansig','tansig','logsig'},'trainlm');

net.trainParam.show = 50;
net.trainParam.lr = 0.5;
net.trainParam.goal = 1e-5;
net.trainParam.epochs = 1000;
[net,tr]=train(net,P,T);
tic
a=sim(net,P)
toc
assignin('base','net',net);
T-a

```

Figura 4.26 Entrenamiento de red neuronal MLP

Para realizar el entrenamiento, se capturo imágenes de diferentes racimos de uva. A cada racimo se tomó 4 fotos en 4 disposiciones distintas, para poder tener más precisión en el entrenamiento. Se realizó el entrenamiento con 60 racimos de uva en diferentes disposiciones

A continuación se muestra el performance de la red, luego de ser entrenada.

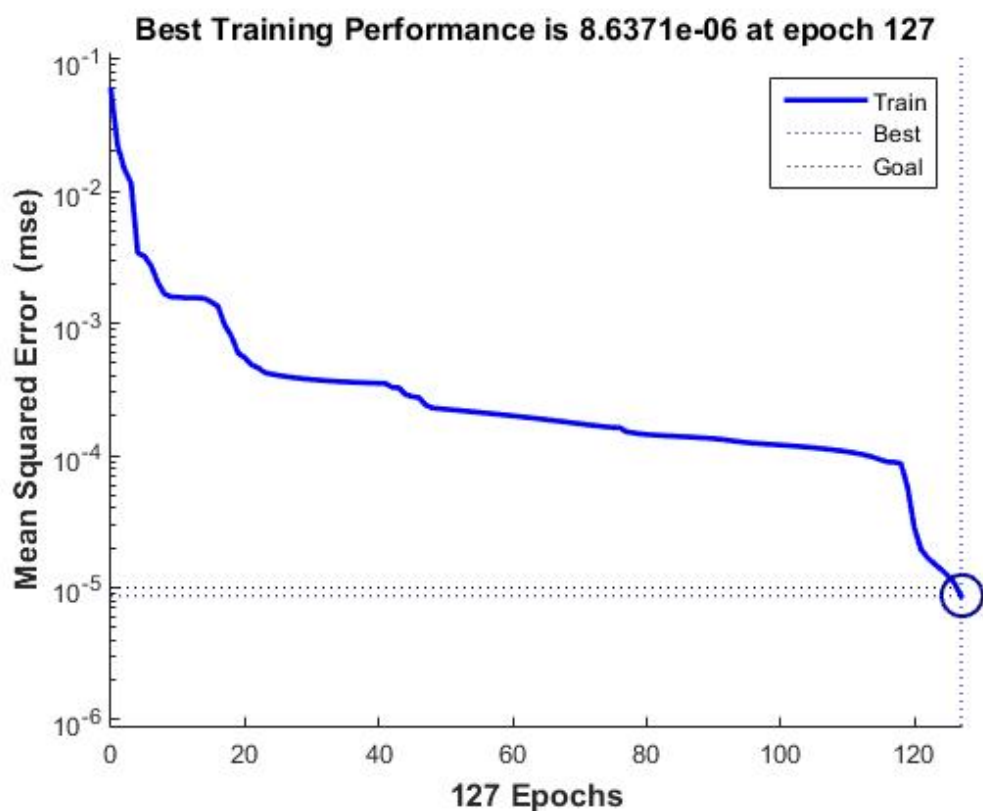


Figura 4.27 Error medio cuadrático de la red

En la figura 4.27, se puede observar el mínimo error cuadrático, que es el error que existe en una época entre la salida de la red neuronal luego del entrenamiento y la salida que se le dio a entrenar al sistema.

En la figura 4.26 se puede observar la línea de código goal, la cual indica un valor de 10-5 esto indica que nuestra fase de entrenamiento se detendrá cuando el error mínimo cuadrático alcance este valor. En la figura anterior se muestra que se alcanza el objetivo requerido

### 4.3.6 Etapa de Pruebas y Resultados

Para realizar estas pruebas se utilizó una interfaz diseñada en Matlab, esta permite la clasificación por peso o tamaño. Esta es la interfaz fue diseñada para permitir una adecuada interacción con el SME. En la siguiente figura se muestra la interfaz diseñada.



Figura 4.28 Interfaz de clasificación de racimos por peso y/o tamaño

Con la red obtenida, luego del entrenamiento se procede a validar los valores con data que no fue entrenada. Al validar los datos, los resultados y los errores obtenidos deben ser del factor de la data entrenada, si hubiera problemas y los errores fueran mayores se debería de entrenar la red modificando la arquitectura o entrenando una vez más. A continuación se muestra una tabla de validación de la data no entrenada.

Racimo de Uva	Disposiciones	Peso Real	Peso Red	Error	Error %
1	1	114	120	6	-5.26
	2		106	-8	7.02
	3		135	21	-18.42
	4		140	26	-22.81
2	1	131	125	-6	4.58
	2		132	1	-0.76
	3		140	9	-6.87
	4		145	14	-10.69
3	1	133.5	142	8.5	-6.37
	2		153	19.5	-14.61
	3		130	-3.5	2.62
	4		118	-15.5	11.61
4	1	233	240	7	-3.00
	2		232	-1	0.43
	3		219	-14	6.01
	4		225	-8	3.43
5	1	260	280	20	-7.69
	2		245	-15	5.77
	3		255	-5	1.92
	4		232	-28	10.77
6	1	134	150	16	-11.94
	2		109	-25	18.66
	3		115	-19	14.18
	4		125	-9	6.72
7	1	406	400	-6	1.48
	2		420	14	-3.45
	3		388	-18	4.43
	4		411	5	-1.23
8	1	220	215	-5	2.27
	2		229	9	-4.09
	3		235	15	-6.82
	4		231	11	-5.00
9	1	350	355	5	-1.43
	2		358	8	-2.29
	3		371	21	-6.00
	4		339	-11	3.14

10	1	204	202	-2	0.98
	2		209	5	-2.45
	3		200	-4	1.96
	4		195	-9	4.41
11	1	451	435	-16	3.55
	2		422	-29	6.43
	3		479	28	-6.21
	4		459	8	-1.77
12	1	206	208	2	-0.97
	2		226	20	-9.71
	3		200	-6	2.91
	4		236	30	-14.56
13	1	704	710	6	-0.85
	2		715	11	-1.56
	3		700	-4	0.57
	4		690	-14	1.99
14	1	727	745	18	-2.48
	2		720	-7	0.96
	3		715	-12	1.65
	4		750	23	-3.16
15	1	253	260	7	-2.77
	2		280	27	-10.67
	3		229	-24	9.49
	4		245	-8	3.16
16	1	189	200	11	-5.82
	2		170	-19	10.05
	3		181	-8	4.23
	4		185	-4	2.12
17	1	687	690	3	-0.44
	2		675	-12	1.75
	3		700	13	-1.89
	4		702	15	-2.18
18	1	215	220	5	-2.33
	2		225	10	-4.65
	3		207	-8	3.72
	4		200	-15	6.98
19	1	494	490	-4	0.81
	2		504	10	-2.02
	3		510	16	-3.24
	4		485	-9	1.82

Tabla 4.6 Validación de red neuronal

El error máximo en la data fue de un error de 53g que solo se dio en una de las imágenes de racimo de uva.

Adicionalmente se realizaron pruebas que permitan clasificar por tamaño de racimo, esto se muestra en la tabla siguiente:

Racimo de Uva	Longitud Real	Longitud extraída del estudio	Error %
1	15	15.3	2.00
2	21	21.5	2.38
3	18	17.4	-3.33
4	12	11.6	-3.33
5	10	10.4	4.00
6	13	13.9	6.92
7	17	17.1	0.59
8	12	12	0.00
9	22	21.8	-0.91
10	14	13.3	-5.00
11	10	10.2	2.00
12	13	13.1	0.77
13	20	20.2	1.00
14	9	8.8	-2.22
15	17	16.9	-0.59
16	25	24.7	-1.20
17	22	22.3	1.36
18	24	23.5	-2.08
19	16	15.4	-3.75
20	14	13.8	-1.43

Tabla 4.7 Validación de algoritmo para extracción de tamaño

#### 4.3.7 Eficiencia de algoritmo desarrollado

Según las tablas mostradas se extrae la eficiencia así como el error máximo de cada control que se requiere tener sobre el racimo de uva como lo son el tamaño y peso.

##### 4.3.7.1 Peso

Observando la tabla 4.6, el error máximo de 50g como peso, y un error en 22.81%. Este error de 50g se manifiesta en racimos cuyos pesos superan los 600g y varían a

diferente disposición. Sería necesario utilizar otra cámara de video que permita obtener el perfil del racimo e ingresar esta data a la red neuronal para ser entrenada, ya que a diferentes disposiciones del racimo el peso de este varía.

#### **4.3.7.2 Tamaño**

Observando la tabla 4.7, el error máximo es de 5% generando así una eficiencia en 95% respecto al tamaño del racimo, no teniendo inconvenientes al variar la disposición del racimo.

#### **4.3.8 Algoritmo de control del sistema mecánico eléctrico**

En esta parte se desarrollara el concepto referente al algoritmo del controlador del sistema mecánico-eléctrico. Se describirá los diagramas de flujo y se adjuntara en el anexo los diagramas escalera de cada diagrama de flujo. Los diagramas de flujo serán del motor eléctrico, la activación del cilindro neumático, la activación del sistema de presurización para limpieza de las uvas de mesa, los sensores de presencia. Las entradas y salidas del controlador fueron mencionadas en la primera parte de este capítulo.

##### **4.3.8.1 Diagrama de Flujo**

Se procede a elaborar el diagrama de flujo de la lógica que controlara el motor eléctrico, este debe de estar protegido contra sobrecarga y cortocircuito.

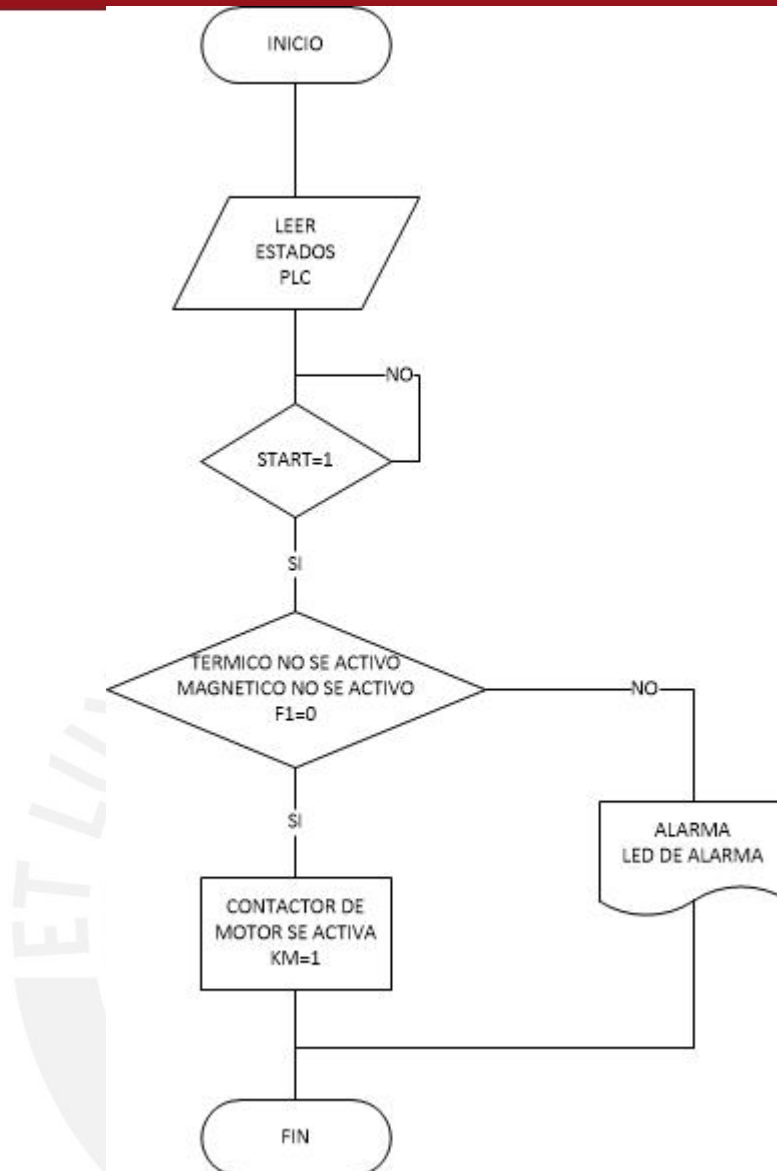


Figura 4.29 Diagrama de flujo control motor M1

Luego se procederá a desarrollar el diagrama de flujo de la señal de mando que activará la bobina de la electroválvula para activar al cilindro neumático para clasificar uvas.

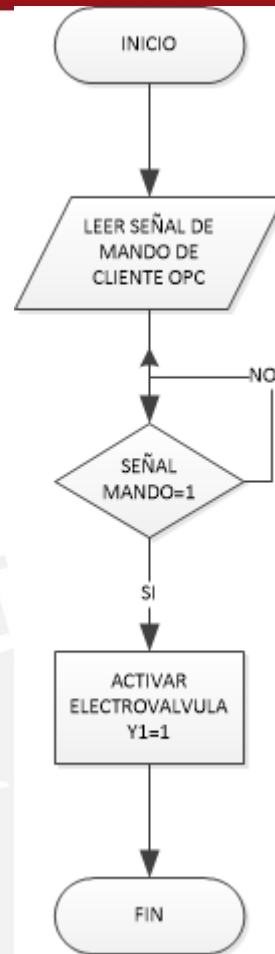


Figura 4.30 Diagrama de Flujo activar Cilindro neumático

Se procederá a mostrar el diagrama de flujo mediante el cual se controlara la limpieza de la uva de mesa.











# ANEXO A

## DIAGRAMA ESCALERA Y CONFIGURACIÓN EN PLC

En este apartado, se muestra la configuración del PLC, así como la programación escalera que se realizó. Para realizar la programación y simulación del programa desarrollado en escalera se utilizó el software del PLC seleccionado el cual se llama TIA PORTAL v12.

