

# Sistema de Visión Artificial para detección de fallas en tela terminada

Jorge Morales<sup>1</sup>, Juan Morales<sup>2</sup>

[jmsegovia@tecnologicoismac.edu.ec](mailto:jmsegovia@tecnologicoismac.edu.ec); [juan.85gms@gmail.com](mailto:juan.85gms@gmail.com)

<sup>1</sup> Instituto Tecnológico Universitario ISMAC, 170184, Quito, Ecuador.

<sup>2</sup> Universidad Politécnica de Valencia, 46022, Valencia, España.

DOI: 10.17013/risti.58.67–80

**Resumen:** La detección de fallas en tela terminada es llevada a cabo mediante la inspección visual que una persona realiza para determinar la afectación directa de diferentes defectos en parte del tejido y así determinar su calidad, tomando como referencia la norma técnica internacional ASTM D 5430-07. Este estudio propone un sistema de Visión Artificial para la detección de las tres principales fallas que se pueden producir en los tejidos como son: hilo sucio, caída de tejido y manchas, mediante cámaras, que en tiempo real encuentran las diferentes fallas mediante algoritmos de detección empleando el programa Python con librerías de cv2. Una vez detectada la falla en la tela, el sistema muestra dónde se encuentra la anomalía mediante la visualización en una pantalla. Para la validación de la detección de las fallas, se realiza la comparación con los métodos: LBP, HAAR y HOG, dando como mejor resultado para la detección de anomalías el LBP.

**Palabras-clave:** Visión Artificial; Tejido; LBP; HAAR; HOG.

## *Artificial Vision System for Failure Detection In Finished Fabric*

**Abstract:** Fault detection in finished fabric is carried out through visual inspection by a person to determine the direct impact of various defects on parts of the fabric and thus determine its quality, referencing the international technical standard ASTM D 5430-07. This study proposes an Artificial Vision system for detecting the three main faults that can occur in fabrics: dirty thread, fabric drop, and stains. Using cameras, the system detects these faults in real-time through detection algorithms employing the Python program with cv2 libraries. Once a fault in the fabric is detected, the system displays the location of the anomaly on a screen. To validate the detection of faults, the comparison is made with the methods: LBP, HAAR and HOG, giving the best result for the detection of anomalies the LBP.

**Keywords:** Artificial Vision, Fabric, LBP, HAAR, HOG.

## 1. Introducción

El proceso de control de calidad de rollos de tela en el área de tejeduría de una planta textil, se basa en la inspección visual y clasificación de tejidos tomando como referencia la norma técnica internacional ASTM D 5430-07 para establecer la afectación directa de diferentes defectos en parte del tejido y así determinar su calidad, todo este proceso lo realiza una persona de manera manual (Standard, 2004), como se muestra en la figura 1.



Figura 1 – Inspección visual de fallas

El tejido terminado puede presentar las siguientes fallas en su fabricación como son: tela apelotonada, caída de tejido, picados, hilo sucio, manchas, líneas verticales, barrado, lascado, falla de diseño, etc (Haider Naqvi et al., 2024).

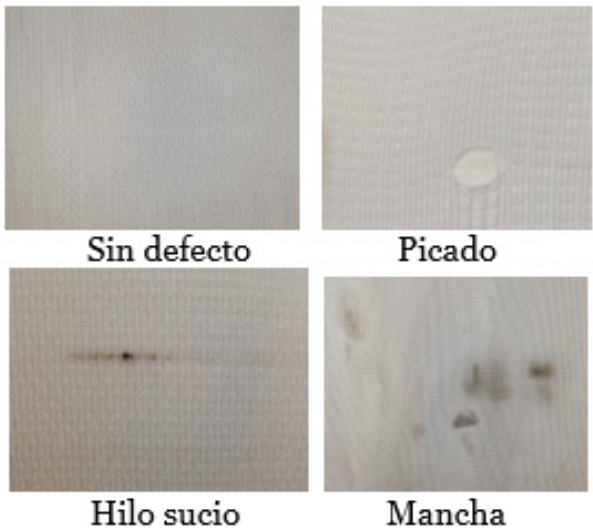


Figura 2 – Ejemplos de fallas en el tejido

La figura 2, muestra las fallas más comunes que se pueden encontrar en el momento de inspección en los tejidos. Estas fallas se generan en el momento de la fabricación de los tejidos, en donde se puede producir un desperfecto en la máquina tejedora circular en diferentes partes de sus mecanismos y generar daños al tejido (Faysal, Hawlader, & Hossain, 2021).

Este trabajo tiene como propósito diseñar un sistema de Visión Artificial que físicamente sea implementado en la máquina de control de calidad, el cual mediante dos cámaras detecte las principales fallas: caída de tejido, manchas e hilo sucio de los rollos de tela terminados. Por lo cual se formula la siguiente pregunta de esta investigación ¿Los sistemas de Visión Artificial son una alternativa para detección de fallas en tela terminada?, que se pretende responder a lo largo del presente este artículo.

## 2. Referencial teórico

La Visión Artificial o también conocida como visión por ordenador es un área científica dentro de la Inteligencia Artificial que se enfoca en el empleo de métodos para adquirir, analizar y tomar decisiones de imágenes del mundo real para generar información numérica para que pueda ser tratada por un computador (Reis et al., 2023), a continuación, se presenta un breve repaso de las técnicas de detección empleadas y trabajos relacionados a esta investigación.

### 2.1. Patrones Locales Binarios (LBP)

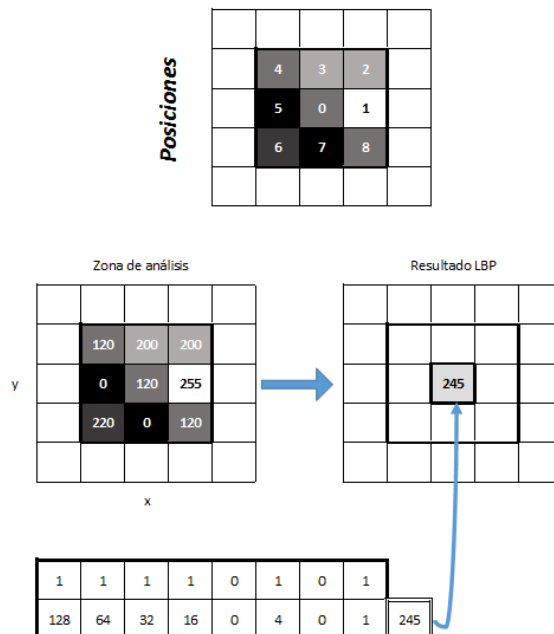


Figura 3 – Funcionamiento del operador LBP

Es una técnica simple pero muy eficiente para la clasificación de objetos con la comparación de sus texturas. Filtra los píxeles adyacentes mediante consideraciones determinadas y obtiene un valor binario representativo. Codifica la relación del píxel central con la intensidad de color de los píxeles cercanos. La propiedad más importante de esta técnica es que es invariante a cambios en los niveles de los colores producidos por cambios en la iluminación, lo que lo convierte en un operador ideal en aplicaciones reales (Huang et al., 2020; Li et al., 2021).

Para aplicar el algoritmo LBP, se debe emplear únicamente un canal de la imagen, por lo general se trabaja en escala de grises o se calcula un LBP por cada canal. Se selecciona un píxel central que es tomado como el umbral con respecto a sus vecinos, se determina un orden de comparación. El funcionamiento del operador LBP se presenta en la figura 3, donde se analiza una ventana de tamaño 3x3 y la relación del píxel central con sus 8 vecinos en este caso. Si el valor del píxel vecino es mayor o igual al valor del píxel central, se asigna un valor de 1 al bit de posición y en caso contrario se asigna un valor de 0. Se completa la comparación con los 8 píxeles adyacentes y se forma un numero binario. En el ejemplo el número binario es 11110101, este valor se transforma a su equivalente decimal que es 245, el cual corresponde al nuevo valor de intensidad de la imagen LBP (Sedaghatjoo et al., 2024).

## 2.2. Histogramas de Gradientes Orientados (HOG)

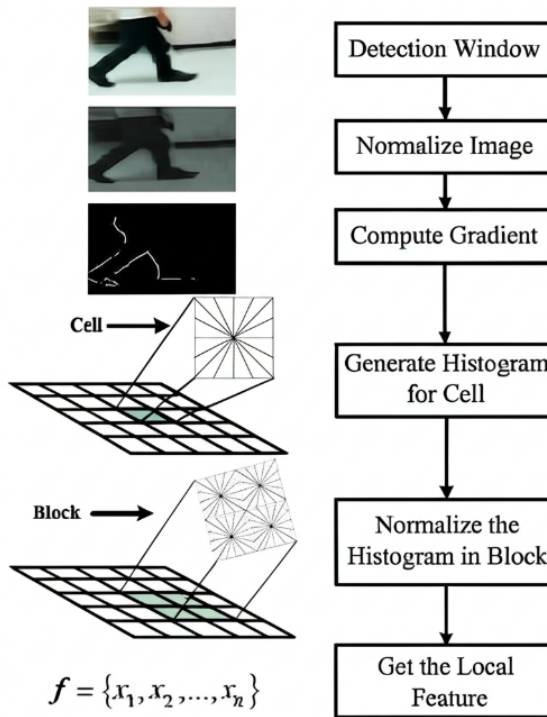


Figura 4 – Histogramas de Gradientes Orientados

Los Histogramas de Gradientes Orientados se basan en la idea de que los objetos pueden ser caracterizados mediante su apariencia. Estos descriptores obtienen la orientación del gradiente correspondiente a cada píxel, obteniendo de esta forma la distribución espacial del objeto.

El descriptor HOG resultante se obtiene a partir del cómputo de histogramas locales de orientación de los gradientes de la imagen, el cuál ha sido previamente dividido en celdas. Por lo tanto, la idea principal de este algoritmo es el diseño de un sistema que divida la imagen en pequeñas regiones llamadas celdas y se obtiene para cada una de ellas un histograma a partir de la orientación de los gradientes de los píxeles que la conforman. Finalmente, la combinación de los histogramas generados para cada una de las celdas proporciona la representación de la imagen, que es representado por el vector de características (Lu et al., 2020).

### 2.3. Clasificador HAAR

Con el clasificador HAAR se pueden detectar diferentes objetos en una imagen. La primera versión de esta técnica fue realizada por (Papageorgiou, 1998), donde se utilizó para la detección de caras y peatones, pero su uso no fue muy significativo hasta unos años más tarde, cuando los investigadores (Viola & Jones, 2004) adaptaron una nueva versión a la que denominaron características Haar. La idea que propone esta técnica consiste en identificar aquellas características de un objeto en base a la estructura de los niveles de intensidad que presentan sus píxeles en una imagen. Esta información es extraída aplicando sobre la imagen del objeto una serie de funciones implementadas bajo las características Haar. La ventaja que presenta esta técnica es que permite detectar la estructura de los objetos, aunque esta no sea uniforme.

Una característica del método Haar es que está definida por diferentes parámetros como el tamaño, la orientación y la distribución de las regiones tanto positivas de color blanco y negativas de color negro. A su vez, estos parámetros dependen de los rasgos del objeto a detectar, y en definitiva de la distribución de intensidades de los píxeles que componen los diferentes rasgos. El objetivo a la hora de construir una característica Haar es buscar que su estructura se asemeje a la del rasgo a detectar tal como bordes, líneas o contornos presentes en una imagen (Bhaidasna et al., 2023)

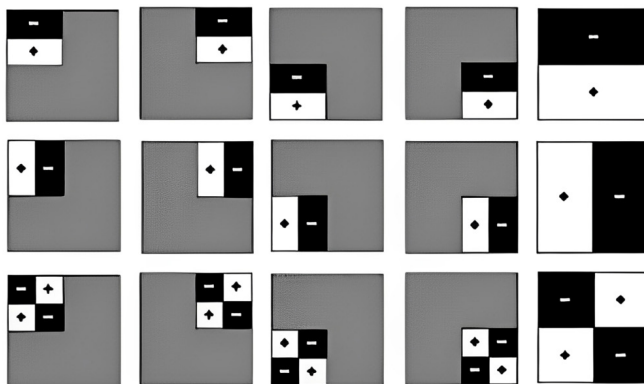


Figura 5 – Distribución de regiones positivas y negativas

En la figura 5, se representa la distribución de forma horizontal, vertical y diagonal que conforman las regiones tanto positivas como negativas del clasificador HAAR.

## 2.4. Trabajos relacionados

Un ejemplo de estudios es el realizado por (Alvarado, 2016) en su publicación: “Detección de defectos en textiles mediante técnicas espaciales para análisis de textura”, en el que busca mejorar el procedimiento de detección de fallas en el tejido que se pueden dar en la máquina de hilar. Generalmente estas fallas son revisadas por operarios visualmente, pero este control es ineficiente y en ocasiones por diversos factores como el cansancio visual o fallas muy pequeñas no se pueden detectar. (Alvarado, 2016) utiliza una base de datos de 2000 imágenes. Estas imágenes fueron proporcionadas por diferentes empresas textiles ubicadas en la ciudad de Ibagué-Colombia, representan diferentes telas que no han sido sometidas a un proceso de tinturado, no contiene estampados, pero presentan diferentes tipos de defectos. Las imágenes se encuentran divididas por sus diferentes defectos en: barrado con 421, motas con 318, piqueos con 431 y sin defecto con 1041.

Otro ejemplo es el trabajo realizado por (Ulloa & Rodríguez 2009) en su estudio: “Detección Automática de Defectos en Telas Basado en la Demodulación AM-FM”. La representación de los componentes FM (Frecuencia Modulada) captura los cambios de los valores de intensidad de la imagen, mientras los componentes AM (Amplitud Modulada) proporcionan información de la energía que se concentra en una determinada región de la imagen.

El procedimiento de detección de defectos se basa en la textura de una determinada región de una imagen que es caracterizada en el histograma de la frecuencia instantánea en el eje vertical (V) y horizontal (H) en dicha región. Luego se divide el resultado de la de demodulación en ventanas pequeñas, cuyo tamaño en píxeles se relaciona con la resolución espacial y se comparan con las distancias con el umbral respectivo, lo cual determina si la ventana analizada representa un defecto (Ulloa & Rodríguez, 2009).

Otro ejemplo es el trabajo presentado por (Banumathi & Nasira 2012) en su estudio: “Fabric Inspection System using Artificial Neural Networks”. En este documento se presenta un nuevo modelo de inspección inteligente de defectos de tejidos.

Emplea imágenes que tienen un tamaño de 300x300 píxeles, con un nivel de gris de 8 bits. El sistema que se plantea adquiere las imágenes de tejido de forma digital, mediante dispositivos de adquisición de imágenes y convierte estas imágenes en formato binario, mediante técnicas de restauración y umbral. Se aplica filtrado en las imágenes para eliminar la saturación de matices que estén presentes en las imágenes. El uso de redes neuronales como clasificador requiere dos fases: una fase de entrenamiento y una fase de prueba. En la fase de entrenamiento, la red neuronal hace el ajuste adecuado para sus pesos (W) para encontrar la respuesta deseada. Cuando la respuesta de salida real es la misma que la requerida, la red ha completado la fase de entrenamiento. En la fase de prueba, la red neuronal realiza la clasificación de un nuevo conjunto de imágenes y su éxito de acierto es evaluado. En este trabajo las redes neuronales fueron entrenadas para detectar y clasificar los defectos de la tela. Los vectores de características se utilizan como los vectores de entrada a la red neuronal. La salida de la imagen procesada se utiliza como una entrada a la red neuronal (NN), que utiliza el algoritmo de backpropagation,

para el cálculo de los factores ponderados y genere la clasificación deseada de defectos como resultado final (Banumathi & Nasira, 2012).

Otro ejemplo de trabajo es el presentado por (Selvi & Nasira 2017) en su estudio: “An effective automatic fabric defect detection system using digital image processing”. En este documento se realiza un clasificador de defectos de tejidos con una red neuronal.

El sistema propuesto inicia con la etapa de preprocesamiento de imagen consiste en una colección de técnicas que se utilizan para mejorar la apariencia de una imagen para su posterior análisis en las etapas siguientes. Para la extracción de características se emplea varios métodos en la captura del contenido específico en las imágenes, con número de características definidas a partir de una imagen. La Clasificación de los defectos da el resultado final de todo el defecto de la tela con el proceso de detección al informar si la tela es aprobada o no. Aplicando redes neuronales como clasificador, se requiere dos fases: una fase de entrenamiento y una fase de prueba. Las imágenes se envían al clasificador de la red neuronal artificial que clasifica las imágenes para que estén defectuosas o sin defectos. Para mejorar el sistema, el clasificador determina exactamente a qué tipo de defecto pertenece el tejido en la imagen. En el sistema propuesto, se han probado varios tipos de defectos para identificar como son: caída de tejido, mancha de tinta, mancha de aceite e hilo perdido.

### **3. Metodología**

Partiendo de un enfoque cuantitativo, con un tipo de investigación experimental y una técnica de levantamiento de datos, donde se tomó como punto de partida un análisis bibliográfico y seguidamente se trabajó con dataset de imágenes de las diferentes fallas en los tejidos. La metodología se basa en la consecución de las siguientes fases: En la primera fase se realizó una revisión bibliográfica de las técnicas empleadas a partir de trabajos relacionados en el tema. La segunda fase fue necesario implementar el Sistema de Visión Artificial en la máquina en donde se realiza la inspección de los tejidos, para lograr tomar las diferentes muestras de los defectos en los tejidos y formar los diferentes dataset de imágenes para su posterior procesamiento, extracción de características e identificación de las tres principales fallas en tiempo real para su validación.

#### **3.1. Capturar la información**

El trabajo comienza con la adecuación del espacio físico de la máquina de control de calidad, para contrarrestar medios externos que afecten con el proceso de captación de imágenes. Para descartar esta posible fuente de ruido se realiza la adecuación lateral de las áreas, para que el fondo pase a ser lo más estable posible durante el proceso de captación de imágenes. Luego se implementa el hardware del sistema tanto para la captación de la información.

La Figura 6 muestra el sistema de hardware implementado, que incluye dos cámaras con una resolución de 720p/30 fps, enfoque fijo y tecnología de lente estándar. Estas cámaras permiten capturar un campo de visión que abarca el ancho de la tela de 1.20 m. Además, se utiliza un monitor para mostrar las imágenes tomadas de las diferentes fallas en la tela terminada durante el control de calidad, lo que permite obtener los dataset de imágenes de cada falla.



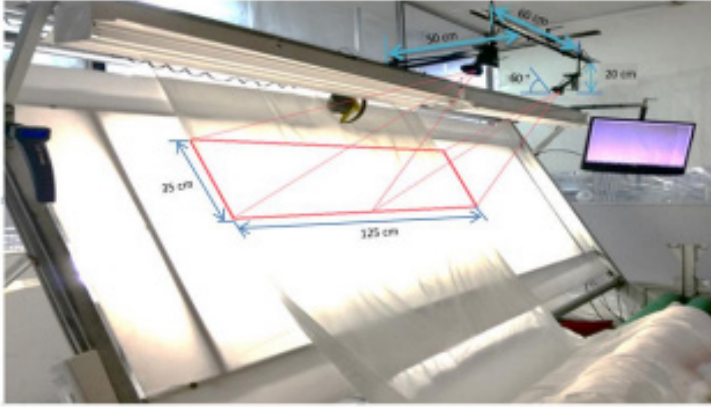


Figura 6 – Sistema de toma de muestras

Con el hardware implementado se realiza la toma de muestras fotográficas de las imágenes para la creación de los dataset de imágenes de los defectos: hilo sucio, caída de tejido y mancha. Cada dataset de imágenes contiene 900 fotografías en formato JPEG con dimensiones de 720 x 720 píxeles.

### 3.2. Transformación de datos

Con el programa Cascade Trainer GUI se transforman los bancos de imágenes de los defectos previamente obtenidos de las tomas de muestras fotográficas en archivos que contienen los clasificadores en cascada de cada defecto en formato XML. Estos archivos son utilizados en el programa principal para la comparación de defectos en tiempo real.

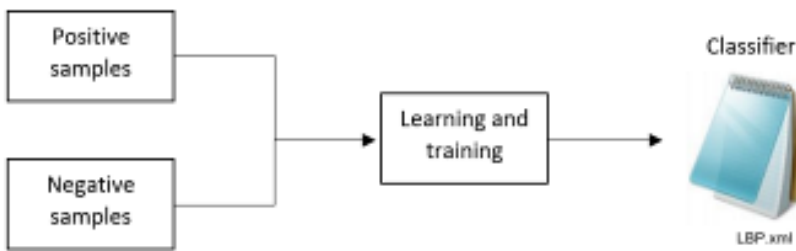


Figura 7 – Diagrama del clasificador

La figura 7 muestra la estructura del clasificador, que a través de la extracción de características de muestras tanto positivas como negativas de imágenes, genera una etapa de aprendizaje y entrenamiento a partir de nuevas imágenes para finalmente obtener un archivo del clasificador de tipo XML.



Para la creación de los clasificadores es necesario crear dos carpetas con da-taset de imágenes. La primera carpeta con el nombre muestras positivas está formada por imágenes que contienen las muestras del banco de imagen del defecto que se desea entrenar para su detección en el proceso de inspección de los rollos de tela.

La segunda carpeta con el nombre muestras negativas, está formada por imágenes en donde no aparecen las fallas de los tejidos, que no son de interés para la detección, estas imágenes son necesarias en la generación del entrena-miento para que el clasificador no genere falsas detecciones. En este caso se colocan imágenes del entorno de la máquina.

El entrenamiento realizado con Cascade Trainer GUI se estructuró en 40 etapas (stages), utilizando 200 clasificadores débiles por etapa (maxWeakCount). El tiempo aproximado requerido para completar el entrenamiento de cada clasificador fue de 5 horas.

### 3.3. Preprocesamiento

Consiste en cambiar de formato RGB a escala de grises del video, para realizar de manera más rápida el procesamiento de la información para las siguientes etapas, se utiliza la función `cv2.cvtColor`.

Para el filtrado de las imágenes que captan las cámaras se utiliza un filtro de suavizado con la función `cv2.GaussianBlur`, que su objetivo en esta etapa es la eliminación del ruido de alta frecuencia que se produce por el paso de la tela por el tablero de inspección de la máquina, en la que los píxeles pueden cambiar rápidamente por el desplazamiento constante de la tela en el proceso de inspección.

### 3.4. Extracción y comparación de características

Para esta etapa se hace uso del clasificador tipo cascada con extension xml. Con el clasificador se realiza la comparación en tiempo real del video captado con las muestras del banco de imágenes con `detectMultiScale` de `cv2`. Si existe una imagen igual que la captada se guardan las coordenadas de la misma para su representación como falla encontrada.

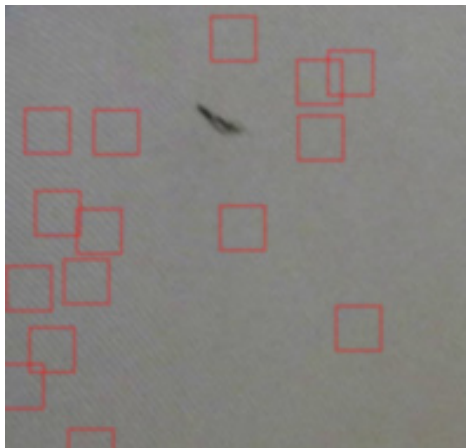


Figura 8 – Comparación de características

En la figura 8, muestra como el programa en tiempo real analiza cuadro a cuadro cambios de textura que se encuentre en el tejido, para así lograr comparar e identificar las fallas en tiempo real.

#### 4. Resultados

Con la ejecución del programa principal, se muestra en pantalla como es la detección de las tres principales fallas en los tejidos. La figura 9, muestra cuando se detecta las tres principales fallas (caída de tejido, hilo sucio, mancha) el sistema automáticamente grafica un recuadro y muestra el tipo de falla que corresponde.



Figura 9 – Fallas detectadas

Para la primera evaluación del sistema se muestra los resultados en tabla 1, que determina el porcentaje de acierto de la detección de fallas, que tienen la técnica empleada en comparación de dos métodos adicionales con referencia al control manual que es llevado a cabo en los inspección de los rollos de tela terminada durante un mes en el control de calidad de alrededor de veinte rollos diarios. En el control manual se detectaron 23 fallas de caída de tejido, 18 fallas en hilo sucio y 15 fallas de mancha, con las técnicas empleadas de detección de fallas se obtiene los porcentajes de acierto que se muestra en la figura 10.

En la figura 10, se muestra que empleando LBP en fallas de caída de tejido se obtuvo un porcentaje de acierto del 92,1%, en fallas de hilo sucio un porcentaje de 93,2% y en manchas un porcentaje de 91,7%. Utilizando HAAR en fallas de caída de tejido se obtuvo un porcentaje de acierto del 90,4%, en fallas de hilo sucio un porcentaje de 89,6% y en manchas un porcentaje de 91,3%. Utilizando HOG en fallas de caída de tejido se obtuvo un porcentaje de acierto del 85,5%, en fallas de hilo sucio un porcentaje de 85,3% y en manchas un porcentaje de 83,4%.

Para la segunda evaluación del sistema se determina el tiempo de procesamiento en segundos que el sistema se toma para la detección de fallas. Este tiempo hace referencia desde que el algoritmo detecta la falla y lo presenta en la pantalla. En la figura 11, se muestra un ejemplo del tiempo que toma el método de Patrones Locales Binarios (LBP) con respecto a los algoritmos: HAAR y HOG en la detección de una misma falla.

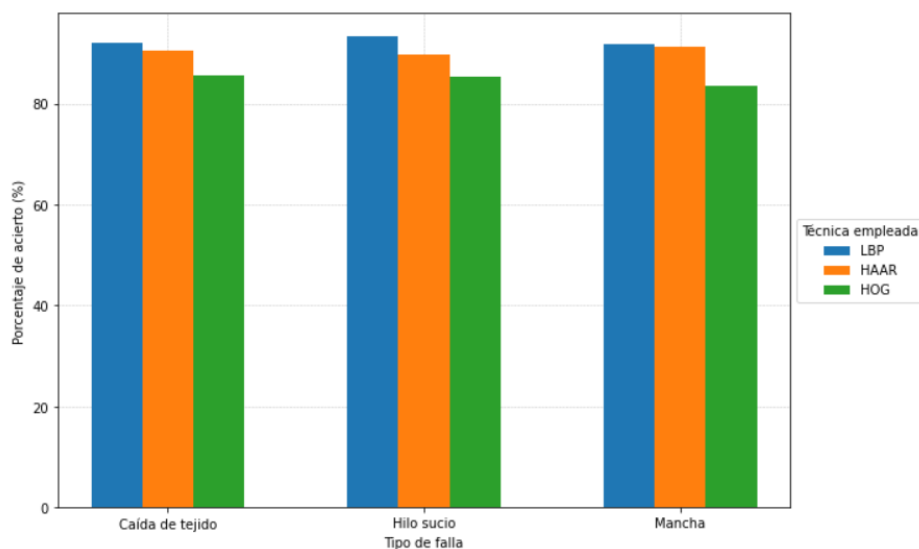


Figura 10 – Porcentaje de acierto por técnica y tipo de falla

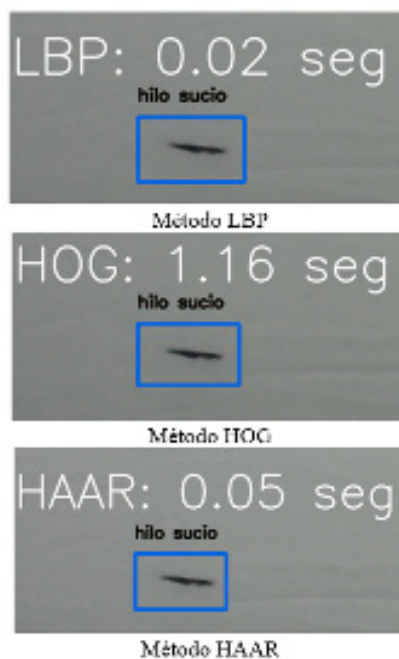


Figura 11 – Tiempos de procesamiento falla: hilo sucio.

La tabla 2, muestra los tiempos de procesamiento en segundos que emplean los algoritmos implementados: LBP, HAAR y HOG en la detección de las fallas: hilo sucio, caída de tejido y mancha. El tiempo de procesamiento es el resultado que les toma a los diferentes métodos en encontrar la falla y mostrar el resultado en pantalla con su etiqueta respectiva.

Fallas	Tiempo de procesamiento de los algoritmos [s]		
	LBP	HAAR	HOG
Hilo sucio	0,02	0,05	1,16
Caída de tejido	0,027	0,063	1,27
Mancha	0,03	0,07	1,35

Tabla 1 – Tiempo de procesamiento.

5. Discusión

La implementación de un sistema de visión artificial para la detección de defectos en tejidos representa un avance significativo en la automatización del control de calidad en la industria textil. A través de la utilización de cámaras y algoritmos de procesamiento de imágenes, se ha logrado no solo identificar fallas de manera más eficiente, sino también reducir la dependencia de la inspección visual humana, que es propensa a errores debido a factores como la fatiga y la variabilidad en la atención.

Los resultados obtenidos en este estudio muestran que el sistema es capaz de detectar con precisión las tres principales fallas: caída de tejido, hilo sucio y manchas. Esto es consistente con los hallazgos de investigaciones previas que han identificado estas fallas como comunes en la producción textil . La capacidad del sistema para clasificar automáticamente los defectos en tiempo real permite una respuesta más rápida en el proceso de producción, lo que puede resultar en una reducción de desperdicios y un aumento en la satisfacción del cliente.

Además, la metodología empleada, que incluye la captura de imágenes en un entorno controlado y la creación de datasets específicos, ha demostrado ser efectiva para entrenar los clasificadores. Esto mejora la precisión del sistema, y también permite la adaptación a diferentes tipos de tejidos y defectos, lo que es importante en un sector tan diverso como el textil.

Sin embargo, es importante considerar algunas limitaciones. La calidad de las imágenes capturadas y la variabilidad en las condiciones de iluminación pueden afectar el rendimiento del sistema. Por lo tanto, futuras investigaciones deberían centrarse en la optimización del hardware y en el desarrollo de algoritmos más robustos que puedan manejar estas variaciones. Además, la inclusión de un mayor número de tipos de defectos en el dataset podría mejorar aún más la capacidad del sistema para generalizar y detectar otro tipos de fallas que se pueden dar en el proceso de fabricación de tela.

## 6. Conclusiones

Con los datos obtenidos del tiempo de procesamiento en los algoritmos empleados, se determina que el algoritmo de Patrones Locales Binarios obtiene un menor tiempo en captar las diferentes fallas y mostrar las mismas en pantalla del sistema. Para las fallas de caída de tejido y mancha, el tiempo de detección aumenta; esto se debe a que este tipo de fallas presentan mayores dimensiones espaciales, lo que aumenta el tiempo de procesamiento para identificar estas fallas.

La recolección de datos para conformar los bancos de imágenes de las diferentes fallas en los tejidos conlleva mucho tiempo realizar la recolección de las imágenes en el proceso de control de calidad, como una propuesta futura es utilizar la técnica de Deep Learning de Data Augmentation que a través de pocas imágenes de muestras de las fallas, permite generar nuevas imágenes y así se puede aumentar el tamaño del conjunto de imágenes que conforman el dataset de cada banco de imágenes para crear el clasificador de textura respectivo.

Para la detección de fallas en tela terminada, el algoritmo de Patrones Locales Binarios, brinda invarianza a la rotación, lo que fue de gran ayuda a la hora de detectar las fallas de la tela en movimiento, lo cual tiene un número de acierto en detección alto, lo que facilita en el momento de detectar y clasificar varias fallas a la vez en un mismo cuadro de visión de las cámaras. Con este tipo de sistemas se podría cambiar el modo de inspección de las fallas de manera automática, mejorando la eficiencia en el proceso de control de calidad de los tejidos con respecto a la forma manual de detección que se emplea.

La implementación de un sistema de Visión Artificial en una máquina de control de calidad no solo permite la detección de fallas en tiempo real, sino que también facilitaría la recolección de datos para el análisis posterior. Esto puede conducir a mejoras continuas en el proceso de producción, ya que se pueden identificar patrones recurrentes de defectos y ajustar los procesos de fabricación de la tela.

## Referencias

- Alvarado Moreno, J. D. (2016). Defect detection in textiles using special techniques fortexure analysis. *Tecnura*, 20(47), 97-106. <https://doi.org/10.14483/udistrital.jour.tecnura.2016>
- Banumathi, P., & Nasira, G. M. (2012). Fabric inspection system using artificial neural networks. *Int. J. Comput. Eng. Sci*, 2(5).
- Bhaidasna, H., & Bhaidasna, Z. (2023). Object Detection Using Machine Learning: A Comprehensive Review. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 9(2), 248–255. <https://doi.org/10.32628/CSEIT2390215>
- Faysal, G. M., Hawlader, M. R., & Hossain, M. M. (2021). Lubrication and Tribological Problem in Textile Industry. *International Journal of Advances Research in Science and Engineering (IJARSE)*.

- Haider Naqvi, S. L., Nadeem, M., Ayub, F., Yasar, A., Zahra Naqvi, S. H., & Tanveer, R. (2024). Social and environmental impacts in textile production. En *SDGs and Textiles* (pp. 423–453). Springer. [https://doi.org/10.1007/978-981-97-5341-3\\_19](https://doi.org/10.1007/978-981-97-5341-3_19)
- Huang, X., Wang, S., & Zhang, Y. (2020). Local binary pattern-based texture analysis for image recognition. *Journal of Imaging*, 6(8), 75.
- Li, J., Zhao, G., & Pietikäinen, M. (2021). Local binary patterns for texture classification: A comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(6), 1985–2005.
- Lu, H., Ma, L., Fu, X., Liu, C., Wang, Z., Tang, M., & Li, N. (2020). Landslides information extraction using object-oriented image analysis paradigm based on deep learning and Transfer Learning. *Remote Sensing*, 12(5), 752. <https://doi.org/10.3390/rs12050752>
- Papageorgiou (1998). A trainable pedestrian detection system. In *Proc. of Intelligent Vehicles* (pp. 241–246).
- Reis, F., Jorge, A., Fernandes, F., & Dias, J. (2023). Protótipo de Solução para Detetar e Sinalizar Defeitos em Pavimentos Rodoviários Baseado em Técnicas de Visión Computacional. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, (52), 25–44. <https://doi.org/10.17013/risti.52.25-44>
- Sedaghatjoo, Z., Hosseinzadeh, H., & Sadeghi Bigham, B. (2024). Local Binary Pattern (LBP) Optimization for Feature Extraction. *arXiv*. <https://arxiv.org/abs/2407.18665>
- Selvi, S. S. T., & Nasira, G. M. (2017). An effective automatic fabric defect detection system using digital image processing. *J. Environ. Nanotechnol*, 6(1), 79–85.
- Standard, A. S. T. M. (2004). Standard Test Methods for Visually Inspecting and Grading Fabrics. ASTM International, 5430(04).
- Ulloa, A., & Rodríguez, P. (2009). Detección automática de defectos en telas basado en la demodulación AM-FM. In *Ibero-American Conference on Trends in Engineering Education and Collaboration*.
- Viola Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*.