

UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO
FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA,
INFORMÁTICA Y MECÁNICA
ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA Y DE
SISTEMAS



TESIS

VISIÓN COMPUTACIONAL APLICADA A LA DETECCIÓN DE
IMPERFECCIONES EN MANTAS DE TEJIDO PUNTO
ELABORADAS CON FIBRAS DE ALPACA

PRESENTADO POR:

Br. ESTEFAN POL SILVA GUEVARA

PARA OPTAR AL TÍTULO PROFESIONAL
DE INGENIERO INFORMÁTICO Y DE
SISTEMAS

ASESOR:

DR. RONY VILLAFUERTE SERNA

CUSCO - PERU
2026



Universidad Nacional de San Antonio Abad del Cusco

INFORME DE SIMILITUD

(Aprobado por Resolución Nro.CU-321-2025-UNSAAC)

El que suscribe, el AsesorDR. Rony Villafuerte Serna.....
..... quien aplica el software de detección de similitud al
trabajo de investigación/tesis titulada: "VISION COMPUTACIONAL APLICADA A
LA DETECCIÓN DE IMPERFECCIONES EN MANTAS DE TEJIDO
PUNTO ELABORADAS CON FIBRA DE ALPACA".....

Presentado por: Estefan Pol Silva Guevara..... DNI N° 73479556;
presentado por: DNI N°:
Para optar el título Profesional/Grado Académico de INGENIERO INFORMATICO
Y DE SISTEMAS.....

Informo que el trabajo de investigación ha sido sometido a revisión por 2..... veces, mediante el
Software de Similitud, conforme al Art. 6° del **Reglamento para Uso del Sistema Detección de**
Similitud en la UNSAAC y de la evaluación de originalidad se tiene un porcentaje de 2.....%.

Evaluación y acciones del reporte de coincidencia para trabajos de investigación conducentes a
grado académico o título profesional, tesis

Porcentaje	Evaluación y Acciones	Marque con una (X)
Del 1 al 10%	No sobrepasa el porcentaje aceptado de similitud.	X
Del 11 al 30 %	Devolver al usuario para las subsanaciones.	
Mayor a 31%	El responsable de la revisión del documento emite un informe al inmediato jerárquico, conforme al reglamento, quien a su vez eleva el informe al Vicerrectorado de Investigación para que tome las acciones correspondientes; Sin perjuicio de las sanciones administrativas que correspondan de acuerdo a Ley.	

Por tanto, en mi condición de Asesor, firmo el presente informe en señal de conformidad y adjunto las primeras páginas del reporte del Sistema de Detección de Similitud.

Cusco, 08 de Agosto..... de 2026.....

Firma

Post firma Dr. Rony Villafuerte Serna

Nro. de DNI 23957778

ORCID del Asesor 0000-0003-4607-522X

Se adjunta:

1. Reporte generado por el Sistema Antiplagio.
2. Enlace del Reporte Generado por el Sistema de Detección de Similitud: oid: 23259: 613183410

Estefan Pol Silva Guevara

TejidoPuntoV3.pdf



Universidad Nacional San Antonio Abad del Cusco

Detalles del documento

Identificador de la entrega

trn:oid:::27259:613183410

Fecha de entrega

8 ago 2026, 6:57 p.m. GMT-5

Fecha de descarga

8 ago 2026, 9:53 p.m. GMT-5

Nombre del archivo

TEjidoPuntoV3.pdf

Tamaño del archivo

7.9 MB

108 páginas

20.103 palabras

112.128 caracteres

2% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe

- Bibliografía
- Texto citado
- Coincidencias menores (menos de 20 palabras)

Exclusiones

- N.º de coincidencias excluidas

Fuentes principales

- 1%  Fuentes de Internet
- 1%  Publicaciones
- 2%  Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.º de alertas de integridad para revisión

No se han detectado manipulaciones de texto sospechosas.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

Dedicatoria

Dedicado a Dios, en primer lugar, por brindarme fortaleza, sabiduría y perseverancia a lo largo de este camino académico.

A mi familia, por su apoyo incondicional, comprensión y confianza constante, quienes han sido mi mayor motivación para seguir adelante.

A mi mama y papa, por su aliento, compañía y apoyo en cada etapa de este proceso, contribuyendo de manera significativa a mi crecimiento personal y profesional.

Agradecimientos

Agradezco a mi asesor de tesis Dr. Rony, por su orientación, paciencia y valiosos aportes académicos que hicieron posible la culminación del presente trabajo.

Asimismo, agradezco a mi familia, a mis padres Paolo y Roxana, a mis hermanos Renzo, Franshesca y Israel, a todos mis familiares y a Beth, por su apoyo constante, comprensión y motivación, los cuales fueron fundamentales para alcanzar este objetivo académico.

Resumen

La industria textil peruana dedicada a productos de fibra de alpaca requiere controles de calidad para identificar imperfecciones en los tejidos. En las mantas de tejido punto, los huecos y puntos corridos se detectan principalmente mediante inspección manual, sujeta a variaciones derivadas del criterio del inspector. Esta investigación tuvo como objetivo implementar un módulo de visión computacional para detectar automáticamente estas imperfecciones.

El estudio fue de enfoque cuantitativo y de tipo aplicado. Se construyó un conjunto de datos propio de 549 imágenes, anotadas en formato YOLO y distribuidas en conjuntos de entrenamiento, validación y prueba. Se entrenaron y compararon tres configuraciones de YOLOv5L con distintas estrategias de optimización, función de pérdida y aumento de datos.

El Experimento 3, basado en aumento de datos avanzado, obtuvo el mejor desempeño en el conjunto de prueba: precisión de 0,983, recall de 0,966, mAP@50 de 0,990 y mAP@50–95 de 0,675. El modelo seleccionado se integró en un prototipo con cámara en vivo que muestra cuadros delimitadores, clases, niveles de confianza y velocidad de procesamiento.

Los resultados evidencian que el módulo permite detectar las imperfecciones estudiadas en un entorno controlado. No obstante, no se evaluaron su desempeño en una línea de producción industrial ni su impacto en los retrabajos, costos y rentabilidad.

Palabras clave: Detección de imperfecciones, Visión computacional, Industria textil, YOLOv5

Abstract

The Peruvian textile industry dedicated to alpaca-fiber products requires quality control processes to identify fabric imperfections. In knitted blankets, holes and dropped stitches are mainly detected through manual inspection, a procedure subject to variations arising from the inspector's judgment. This research aimed to implement a computer vision module to automatically detect these imperfections.

The study followed a quantitative and applied approach. A custom dataset comprising 549 images was created, annotated in YOLO format, and divided into training, validation, and test sets. Three YOLOv5L model configurations were trained and compared using different optimization, loss-function, and data-augmentation strategies.

Experiment 3, based on advanced data augmentation, achieved the best performance on the test set: a precision of 0.983, a recall of 0.966, an mAP@50 of 0.990, and an mAP@50–95 of 0.675. The selected model was integrated into a live-camera prototype that displays bounding boxes, classes, confidence scores, and processing speed.

The results show that the module can detect the studied imperfections in a controlled environment. However, neither its performance on an industrial production line nor its impact on rework, costs, and profitability was evaluated.

Keywords: Imperfections detection, Computer vision, Textile industry, YOLOv5

Introducción

La industria textil peruana, particularmente el sector dedicado a la producción de mantas de tejido punto elaboradas con fibras de alpaca, enfrenta importantes desafíos en el ámbito del control de calidad. La detección de imperfecciones como huecos y puntos corridos continúa realizándose, en muchos casos, mediante inspección visual manual, lo que introduce subjetividad, errores humanos y limitaciones en la eficiencia del proceso productivo. En este contexto, la aplicación de técnicas de visión computacional y aprendizaje profundo representa una alternativa tecnológica con alto potencial para automatizar y optimizar los procesos de inspección.

La presente tesis tiene como objetivo implementar un módulo de visión computacional para la detección automática de imperfecciones en mantas de tejido punto de alpaca, empleando modelos de detección de objetos basados en la arquitectura YOLOv5. Para ello, se desarrolló un conjunto de datos propio, se entrenaron y evaluaron distintos modelos, y se integró la solución en un prototipo capaz de operar en tiempo real, comprobando su funcionamiento mediante una cámara en vivo bajo condiciones controladas.

El documento se encuentra estructurado en cuatro capítulos, los cuales se describen a continuación:

El **Capítulo 1: Aspectos generales** presenta el planteamiento del problema, la justificación de la investigación, los objetivos generales y específicos, así como el alcance y las limitaciones del estudio. En este capítulo se contextualiza la problemática del control de cali-

dad en la industria textil de alpaca y se fundamenta la necesidad de incorporar herramientas tecnológicas basadas en inteligencia artificial.

El **Capítulo 2: Marco teórico** desarrolla los fundamentos conceptuales que sustentan la investigación. Se abordan temas relacionados con la industria textil de alpaca, las imperfecciones en tejidos de punto, los principios de la visión computacional, las redes neuronales convolucionales, el aprendizaje profundo y las arquitecturas de detección de objetos, con énfasis en el modelo YOLOv5. Asimismo, se presentan los antecedentes nacionales e internacionales relevantes para el estudio.

El **Capítulo 3: Desarrollo del proyecto** describe la metodología empleada para la implementación del prototipo propuesto. En este capítulo se detalla la construcción del dataset de mantas de alpaca, el proceso de etiquetado de imperfecciones, las configuraciones de entrenamiento, el uso de técnicas de aumento de datos, y la evaluación de distintos experimentos hasta la selección del modelo con mejor desempeño. Además, se explica la integración del modelo entrenado en un entorno de ejecución en tiempo real mediante Python, utilizando librerías como PyTorch y OpenCV, permitiendo la detección de imperfecciones en secuencias de video en vivo con visualización de cuadros delimitadores, niveles de confianza y velocidad de procesamiento en 40 cuadros por segundo (FPS).

Finalmente, El **Capítulo 4: Análisis y discusión de resultados** presenta el análisis de los resultados obtenidos en relación con los objetivos planteados, considerando métricas como precisión, recall, mAP@50 y mAP@50–95. Asimismo, se desarrolla la discusión comparativa con los antecedentes revisados. Posteriormente, se presentan las conclusiones derivadas de la investigación y las recomendaciones para trabajos futuros.

En conjunto, la estructura de la tesis permite abordar de manera integral la problemática del control de calidad en mantas de tejido punto de alpaca, desde su fundamentación teórica hasta la validación práctica de una solución basada en visión computacional.

Listado de Abreviaturas

CNN	Convolutional Neural Network (Red Neuronal Convolutional)
YOLO	You Only Look Once
YOLOv5	You Only Look Once versión 5
mAP	mean Average Precision (Precisión Promedio Media)
mAP@50	mean Average Precision con umbral de IoU = 0.50
mAP@50–95	mean Average Precision con umbrales de IoU de 0.50 a 0.95
FPS	Frames Per Second (Cuadros por segundo)
TP	True Positive (Verdadero Positivo)
FP	False Positive (Falso Positivo)
FN	False Negative (Falso Negativo)
F1	F1-Score (Puntuación F1)
CV	Computer Vision (Visión Computacional)
LR	Learning Rate (Tasa de aprendizaje)
SGD	Stochastic Gradient Descent (Descenso de Gradiente Estocástico)
AdamW	Adam with Weight Decay (Adam con Decaimiento de Peso)

Índice general

Dedicatoria	II
Agradecimientos	III
Resumen	IV
Abstract	V
Introducción	VI
Listado de Abreviaturas	VIII
Índice General	IX
Índice de figuras	XIV
Índice de tablas	XVI
Índice de códigos	XVII
1. Aspectos Generales	1
1.1. Planteamiento del Problema	1
1.1.1. Descripción del problema	1
1.1.2. Identificación del problema	2
1.2. Formulación del Problema	4

1.2.1.	Problema General	4
1.2.2.	Problemas Específicos	4
1.3.	Objetivos	5
1.3.1.	Objetivo General	5
1.3.2.	Objetivos Específicos	5
1.4.	Justificación	6
1.4.1.	Conveniencia	6
1.4.2.	Relevancia	6
1.4.3.	Implicancias Prácticas	6
1.4.4.	Valor Teórico	7
1.4.5.	Utilidad Metodológica	7
1.5.	Delimitación de estudio	8
1.5.1.	Delimitación Espacial	8
1.5.2.	Delimitación Temporal	8
1.6.	Método de desarrollo	8
1.6.1.	Enfoque	8
1.6.2.	Alcance	9
1.6.3.	Diseño	9
1.6.4.	Cronograma de Actividades	10
2.	Marco Teórico	12
2.1.	Antecedentes	12
2.1.1.	Antecedentes Internacionales	12
2.2.	Bases Teóricas	16
2.2.1.	Fibra de Alpaca	16
2.2.2.	Procesamiento de la Fibra de Alpaca	17

2.2.3.	Técnicas de Tejido Punto	18
2.2.4.	Visión Artificial	21
2.2.5.	Imagen Digital	22
2.2.6.	Procesamiento de Imágenes Digitales	23
2.2.7.	Aprendizaje de Máquina	23
2.2.8.	Aprendizaje Profundo	24
2.2.9.	Redes Neuronales Artificiales	24
2.2.10.	Redes Neuronales Convolucionales	24
2.2.11.	Optimizadores en Redes Neuronales Profundas	26
2.2.12.	Métricas para la evaluación de modelos de detección	26
2.2.13.	Técnicas de Evaluación de Modelos	29
2.2.14.	Transferencia de Aprendizaje	30
2.2.15.	Data Augmentation	31
2.2.16.	Arquitecturas de Redes Neuronales Convolucionales	35
2.2.17.	YOLO	35
3.	Desarrollo del tema de tesis	36
3.1.	Metodología	36
3.2.	Recolección y construcción del conjunto de datos	37
3.2.1.	Definición de clases	37
3.2.2.	Proceso de captura de imágenes	38
3.2.3.	Anotación y composición del conjunto de datos	38
3.2.4.	División y organización del conjunto de datos	39
3.2.5.	Publicación y disponibilidad del conjunto de datos	40
3.3.	Preprocesamiento de imágenes	40
3.3.1.	Evaluación y selección de filtros de preprocesamiento	41

3.3.2.	Evaluación de transformaciones para el aumento de datos	41
3.4.	Justificación de la arquitectura YOLOv5L	44
3.5.	Entrenamiento de modelos	45
3.5.1.	Definición de la arquitectura YOLOv5L	45
3.5.2.	Diseño experimental	46
3.5.3.	Entorno y procedimiento de entrenamiento	46
3.5.4.	Configuración común de entrenamiento	47
3.5.5.	Configuraciones experimentales	49
3.5.6.	Criterio de evaluación y selección	52
3.6.	Resultados de la evaluación experimental	52
3.6.1.	Comparación en el conjunto de validación	52
3.6.2.	Selección del modelo con mejor desempeño	53
3.6.3.	Comparación en el conjunto de prueba	54
3.6.4.	Comparación global de los experimentos	56
3.6.5.	Análisis del modelo seleccionado	57
3.6.6.	Resultados visuales del modelo seleccionado	59
3.6.7.	Almacenamiento e integración del modelo final	61
3.7.	Implementación del prototipo de detección en tiempo real	62
3.7.1.	Flujo de desarrollo del prototipo	62
3.8.	Limitaciones del estudio	66
4.	Análisis y discusión de resultados	68
4.1.	Análisis de resultados respecto a los objetivos	68
4.2.	Discusión de resultados respecto a los antecedentes	70
	Conclusiones	73

Recomendaciones	75
Bibliografía	77
Anexos	82
A. Código de entrenamiento del Experimento 3 en Google Colab	82
B. Código de implementación del prototipo de detección en tiempo real	85

Índice de figuras

1.1. <i>Diagrama de Gantt elaborado en Lucidchart.</i>	11
2.1. Comparación visual de fibra de alpaca.	17
2.2. Análisis del ciclo de vida fibra de alpaca.	18
2.3. Tejido en Punto Jersey.	19
2.4. Tejido en Punto Jacquard.	20
2.5. Tejido en Punto Intarsia.	21
2.6. Imagen digital	22
2.7. Aumento de datos mediante volteo	32
2.8. Aumento de datos mediante Rotacion	32
2.9. aumento de datos mediante Escalado	33
2.10. aumento de datos mediante Recorte	34
2.11. aumento de datos mediante Traslación	34
3.1. Matriz de confusión del Experimento 3 en el conjunto de prueba	57
3.2. Curva Precision–Recall del Experimento 3 en el conjunto de prueba	58
3.3. Curva F1–Confidence del Experimento 3 en el conjunto de prueba	59
3.4. Anotaciones de referencia de una muestra del conjunto de validación	60
3.5. Predicciones del Experimento 3 sobre una muestra del conjunto de validación	61
3.6. Diagrama de flujo del desarrollo del prototipo de detección en tiempo real. .	63

3.7. Captura en tiempo real de imperfecciones en mantas de alpaca	65
---	----

Índice de tablas

3.1. Distribución del conjunto de datos	39
3.2. Filtros analizados durante el preprocesamiento de imágenes	42
3.3. Transformaciones analizadas para el aumento de datos	43
3.4. Configuraciones evaluadas durante el entrenamiento	46
3.5. Configuración común utilizada en los cuatro experimentos	48
3.6. Configuración específica del Experimento 1	49
3.7. Configuración específica del Experimento 2	50
3.8. Configuración específica del Experimento 3	51
3.9. Configuración específica del Experimento 4	51
3.10. Comparación de los experimentos en el conjunto de validación	53
3.11. Resultados del Experimento 1 en el conjunto de prueba	54
3.12. Resultados del Experimento 2 en el conjunto de prueba	54
3.13. Resultados del Experimento 3 en el conjunto de prueba	55
3.14. Resultados del Experimento 4 en el conjunto de prueba	55
3.15. Comparación promedio de los experimentos en el conjunto de prueba	55
3.16. Comparación global de los experimentos	56

Índice de códigos

3.1. Configuración del conjunto de datos en formato YOLO	39
A.1. Código del Experimento 3: YOLOv5L con aumento de datos avanzado	82
B.1. Código del prototipo de detección en tiempo real	85

Capítulo 1

Aspectos Generales

1.1. Planteamiento del Problema

1.1.1. Descripción del problema

A nivel mundial, la industria textil enfrenta desafíos significativos en la modernización de sus procesos de control de calidad. La producción de mantas de tejido punto elaboradas con fibras de alpaca exige métodos de inspección más precisos y eficientes, debido a las particularidades de este material y a las altas exigencias de calidad que demandan los mercados competitivos. Investigaciones recientes han demostrado que el uso de técnicas basadas en aprendizaje profundo puede mejorar de forma notable la identificación de imperfecciones en tejidos con patrones complejos, como los desarrollados mediante procesos Jacquard o warp-knitted, logrando una mayor discriminación entre áreas defectuosas y no defectuosas mediante modelos como autoencoders apilados con criterio de Fisher Senthilkumar et al. [2024].

En América Latina, se observa una creciente adopción de tecnologías avanzadas en la industria textil, lo que ha resultado clave para incrementar la competitividad del sector.

En Colombia, por ejemplo, la incorporación de sistemas de automatización ha permitido a las empresas adaptarse a las demandas cambiantes y mejorar su presencia en el mercado Lopera Lopera and Velez-Ocampo [2021]. Asimismo, la región avanza hacia modelos de economía circular y sostenible, buscando reducir el impacto ambiental y promover prácticas responsables García Jarpa and Halog [2021].

En el Perú, la industria textil dedicada a la producción de mantas de tejido de punto enfrenta una problemática relacionada con la carencia de un adecuado control de calidad. Esta carencia se traduce en altos índices de imperfecciones, retrabajos y pérdidas de material, afectando la rentabilidad y la competitividad. La implementación de sistemas de inspección mediante visión computacional y algoritmos de aprendizaje automático, tiene el potencial de aumentar la precisión y la velocidad de las inspecciones, reduciendo costos y mejorando la calidad final del producto Kumuthini et al. [2023].

1.1.2. Identificación del problema

El presente estudio se focaliza en el sector textil dedicado a la producción de mantas de tejido punto elaboradas con fibras de alpaca. En este tipo de manufactura, se realizan múltiples controles de calidad a lo largo del proceso productivo para garantizar que el producto final cumpla con los estándares establecidos.

El primer control de calidad se lleva a cabo cuando se confeccionan las piezas individuales de la manta o durante la producción inicial del tejido. En esta etapa, el personal verifica si se presentan imperfecciones antes de pasar a procesos posteriores como el ensamblaje, unión de piezas o acabado.

El segundo control se realiza una vez que el tejido ha pasado por procesos intermedios, como ajustes o acabados parciales. En esta fase se inspecciona nuevamente la manta con el objetivo de detectar posibles imperfecciones originadas durante el proceso productivo.

Finalmente, se efectúa el tercer y último control de calidad, que constituye el enfoque de esta investigación. Este control se realiza después del lavado, planchado o procesos de acabado final, y consiste en una inspección visual detallada por parte del personal especializado. En esta etapa se identifican imperfecciones de manufactura como huecos y puntos corridos, los cuales afectan directamente la calidad y presentación del producto final.

Entre las imperfecciones más comunes que se presentan en las mantas de tejido punto elaboradas con fibras de alpaca se encuentran los huecos y los puntos corridos. Estas imperfecciones se originan durante el proceso de tejido. Si las imperfecciones son pequeñas, el personal encargado las corrige manualmente utilizando una aguja; sin embargo, cuando el daño es considerable y no puede ser reparado, la manta es desechada, generando pérdidas económicas y desperdicio de material. Este procedimiento manual, aunque tradicionalmente utilizado, presenta una serie de limitaciones: la detección de imperfecciones depende de la experiencia y atención del operario, lo que introduce un alto grado de subjetividad y probabilidad de error. Asimismo, el proceso es lento y no escalable, especialmente en contextos donde se busca aumentar la productividad o estandarizar la calidad.

Las limitaciones del control de calidad basado principalmente en la inspección visual manual pueden ocasionar que algunas imperfecciones no sean identificadas oportunamente, generando retrabajos, desperdicio de materiales y posibles pérdidas económicas. Asimismo, la dependencia de la experiencia y atención del operario introduce subjetividad y dificulta

la estandarización del proceso de inspección final.

En respuesta a esta problemática, se propone la implementación de un prototipo basado en visión computacional, cuyo principal aporte sería la detección de huecos y puntos corridos en mantas de tejido punto elaboradas con fibras de alpaca. Se espera que la implementación del prototipo contribuya a reducir errores, disminuir retrabajos, optimizar el uso de materiales y mejorar la eficiencia general del proceso productivo, impulsando así la modernización del sector textil en el Perú.

1.2. Formulación del Problema

1.2.1. Problema General

¿Cómo implementar un módulo de visión computacional basado en YOLOv5 para la detección de imperfecciones (huecos y puntos corridos) en mantas de tejido punto elaboradas con fibras de alpaca, y qué desempeño alcanza mediante las métricas de precisión, recall y mAP@50?

1.2.2. Problemas Específicos

- ¿Cómo construir un conjunto de datos de imágenes de mantas de tejido punto elaboradas con fibras de alpaca que permita entrenar y validar modelos de visión computacional?
- ¿Qué nivel de precisión y capacidad de detección alcanzan los modelos de visión computacional desarrollados para identificar imperfecciones de manufactura (huecos y puntos corridos) en mantas de tejido punto de alpaca?

- ¿Cómo integrar el modelo de visión computacional seleccionado en un prototipo que permita detectar huecos y puntos corridos mediante una cámara en vivo durante la etapa de inspección visual final de las mantas de tejido punto elaboradas con fibras de alpaca?

1.3. Objetivos

1.3.1. Objetivo General

Implementar un módulo de visión computacional basado en YOLOv5 para la detección de imperfecciones (huecos y puntos corridos) en mantas de tejido punto elaboradas con fibras de alpaca, logrando un desempeño evaluado mediante métricas como precisión, recall y mAP@50.

1.3.2. Objetivos Específicos

- Construir un conjunto de datos de imágenes de mantas de tejido punto elaboradas con fibras de alpaca, que sirva para entrenar y validar modelos de visión computacional.
- Desarrollar modelos de visión computacional que permitan identificar imperfecciones de manufactura en las mantas de tejido punto, tales como huecos y puntos corridos.
- Integrar el modelo de visión computacional seleccionado en un prototipo de detección mediante cámara en vivo para apoyar la identificación de huecos y puntos corridos durante la etapa de inspección visual final de las mantas de tejido punto elaboradas con fibras de alpaca.

1.4. Justificación

1.4.1. Conveniencia

El estudio resulta conveniente debido a que la industria textil dedicada a la elaboración de mantas de tejido punto con fibras de alpaca continúa utilizando métodos manuales para el control de calidad, los cuales presentan limitaciones en términos de precisión, rapidez y estandarización. Estas deficiencias generan retrabajos, desperdicio de materia prima y pérdidas económicas. Investigar la aplicación de técnicas de visión computacional permite abordar esta problemática mediante el uso de tecnologías actuales que optimicen el proceso de inspección, contribuyendo a la mejora de la eficiencia productiva y a la reducción de errores humanos.

1.4.2. Relevancia

La investigación es relevante porque responde a una necesidad real del sector textil peruano, donde existe una demanda creciente por productos de alta calidad que cumplan con estándares competitivos a nivel nacional e internacional. La detección automatizada de imperfecciones como huecos y puntos corridos en mantas de alpaca permitirá mejorar el control de calidad del producto final, incrementando la confianza del cliente y fortaleciendo la competitividad de las empresas textiles. Asimismo, el estudio promueve la adopción de tecnologías de transformación digital en un sector tradicional de la economía.

1.4.3. Implicancias Prácticas

Los resultados de esta investigación podrán ser aplicados directamente en entornos productivos reales mediante la implementación de un prototipo de control de calidad basado en visión computacional. Dicho prototipo podrá ser utilizado por el personal encargado

de la inspección final de las mantas, facilitando la detección temprana de imperfecciones, reduciendo retrabajos y optimizando el uso de materiales. A futuro, el prototipo podrá adaptarse o ampliarse para otros tipos de tejidos o productos textiles, contribuyendo a la modernización del proceso productivo.

1.4.4. Valor Teórico

Desde el punto de vista teórico, la investigación aporta al cuerpo de conocimiento científico relacionado con la aplicación de la visión computacional y el aprendizaje automático en la industria textil, específicamente en productos elaborados con fibras de alpaca. La construcción de un conjunto de datos especializado y el desarrollo de modelos para la detección de imperfecciones en tejidos de punto representan un aporte relevante, considerando la limitada literatura existente en este ámbito. Los resultados obtenidos podrán servir como referencia para futuras investigaciones académicas.

1.4.5. Utilidad Metodológica

La metodología propuesta en esta investigación podrá ser reutilizada en estudios similares que busquen aplicar técnicas de visión computacional en procesos de control de calidad industrial. El enfoque empleado para la adquisición de imágenes, construcción del conjunto de datos, preprocesamiento, entrenamiento y validación de modelos puede adaptarse a otros tipos de materiales o imperfecciones. De esta manera, el trabajo metodológico desarrollado constituye una base replicable y escalable para investigaciones futuras.

1.5. Delimitación de estudio

1.5.1. Delimitación Espacial

El presente estudio se desarrolla en el sector textil dedicado a la producción de mantas de tejido punto elaboradas con fibras de alpaca. De manera específica, una parte del estudio se realiza en la empresa APU KUNTUR CUSCO, ubicada en la región Cusco, donde se lleva a cabo el proceso productivo y el control de calidad de las mantas. En este entorno se obtienen las imágenes necesarias para la construcción del conjunto de datos y se analizan las condiciones reales de producción para la aplicación del prototipo de visión computacional propuesto.

1.5.2. Delimitación Temporal

La investigación se desarrolla durante el periodo correspondiente al año 2025–2026, comprendiendo las etapas de recolección de datos, desarrollo del conjunto de imágenes, diseño y entrenamiento de los modelos de visión computacional. Las actividades del estudio se realizan de manera progresiva conforme al cronograma establecido para el desarrollo de la tesis.

1.6. Método de desarrollo

1.6.1. Enfoque

La investigación en esta tesis es de enfoque cuantitativo, conforme a la definición de Hernández-Sampieri et al. [2014]. Este tipo de estudio se centra en la observación y descripción detallada de características, comportamientos o fenómenos tal como ocurren. En este caso, el objetivo principal es evaluar y comparar el rendimiento de diferentes modelos de

visión computacional, para la detección de imperfecciones textiles, como huecos y puntos corridos. Se fundamenta en la necesidad de ofrecer una caracterización detallada del desempeño de modelos de visión computacional seleccionados y métodos de procesamiento de imágenes. Este análisis se llevará a cabo en un entorno controlado, utilizando conjuntos de datos creados para este propósito y bajo condiciones estandarizadas, con el fin de garantizar una evaluación objetiva y confiable.

1.6.2. Alcance

La presente investigación tiene un alcance descriptivo. Este tipo de alcance se orienta a describir y analizar el comportamiento de un fenómeno tal como se presenta en la realidad, sin establecer relaciones de causalidad entre variables. En el contexto del estudio, se describe el desempeño de modelos de visión computacional aplicados a la detección de imperfecciones en mantas de tejido punto elaboradas con fibras de alpaca, específicamente huecos y puntos corridos. El análisis se centra en la evaluación cuantitativa de los resultados obtenidos por los modelos, permitiendo caracterizar su precisión y efectividad dentro del proceso de control de calidad.

1.6.3. Diseño

La presente investigación adopta un diseño experimental de tipo aplicado, debido a que se manipulan variables relacionadas con la configuración y entrenamiento de modelos de visión computacional con el fin de evaluar su desempeño en la detección de imperfecciones textiles. Según Hernández-Sampieri et al. [2014], los estudios experimentales se caracterizan por la manipulación intencional de variables para analizar sus efectos en otras.

Asimismo, el estudio es de corte transversal, ya que la recolección y análisis de datos se realiza en un único periodo de tiempo, sin seguimiento longitudinal. El diseño experimental

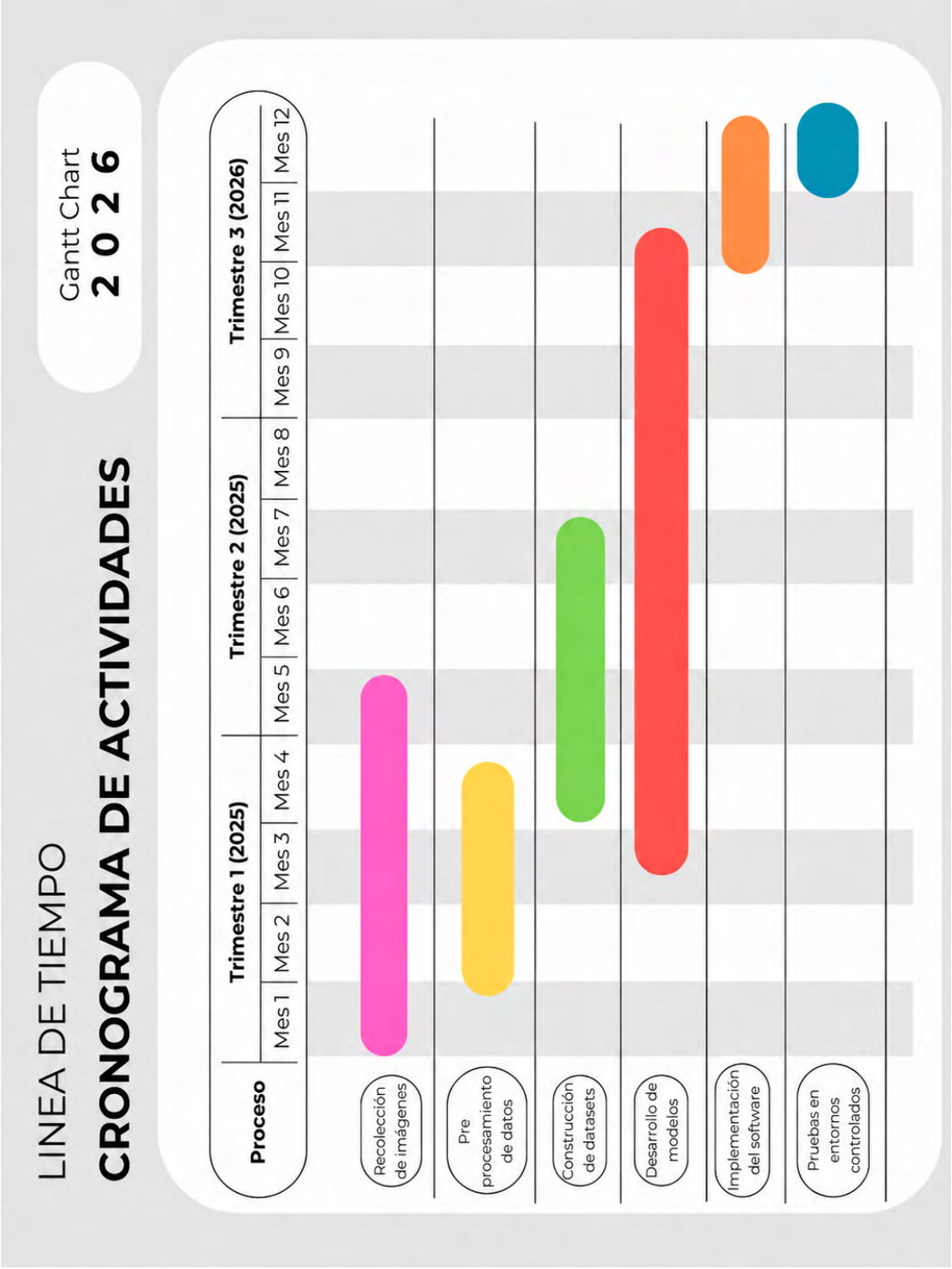
se justifica en la necesidad de comparar diferentes configuraciones del modelo YOLOv5, evaluando su impacto en métricas de desempeño como precisión, recall y mAP@50.

En este contexto, la variable independiente está representada por la implementación y configuración del modelo de visión computacional basado en YOLOv5, mientras que la variable dependiente corresponde al desempeño en la detección de imperfecciones (huecos y puntos corridos) en mantas de tejido punto de alpaca, medido a través de indicadores cuantitativos.

Este diseño permite establecer una evaluación objetiva del comportamiento de los modelos desarrollados en condiciones controladas, garantizando la reproducibilidad y validez de los resultados obtenidos.

1.6.4. Cronograma de Actividades

Figura 1.1
Diagrama de Gantt elaborado en Lucidchart.



Capítulo 2

Marco Teórico

2.1. Antecedentes

2.1.1. Antecedentes Internacionales

Hoyos Montes [2020]. *Detección de defectos en fibras textiles utilizando algoritmos de deep learning* (Informe técnico). Universidad de Antioquia, Medellín, Colombia. La fuente de datos fue el conjunto TexUAN, compuesto por imágenes proporcionadas por empresas textiles de Ibagué, departamento de Tolima, Colombia.

Conclusiones:

- Se implementó un sistema de detección de defectos textiles basado en técnicas de procesamiento de imágenes y algoritmos de deep learning, integrados en un servicio web.
- El sistema empleó Google Cloud AutoML Vision y un dataset especializado denominado TexJAN para entrenar y validar el modelo.
- Alcanzó una precisión del 89.89 % en la clasificación y una efectividad del 77.5 % en la

detección de defectos.

- El estudio concluyó que este tipo de soluciones tiene un alto potencial para mejorar la competitividad en la industria textil al automatizar procesos que tradicionalmente son manuales.

Comentario: Este informe será fundamental para guiar la implementación de un sistema de control de calidad automatizado basado en deep learning, especialmente por el uso de servicios en la nube para el entrenamiento de modelos.

Zhang [2021]. *Cascaded zoom-in network for patterned fabric defect detection*, arXiv preprint arXiv:2108.06760. La investigación empleó conjuntos de imágenes reales de tejidos estampados; sin embargo, el documento consultado no especifica de manera explícita la región ni el país donde fueron capturadas las imágenes.

Conclusiones:

- Se propuso una red de dos pasos (CZI-Net) combinando A-HOG y SIFT para detectar imperfecciones en telas con patrones complejos.
- El enfoque incluía un método de extracción de características eficiente (BoI, LCRE, RLC) que mejoró la precisión y redujo el costo computacional.
- Obtuvo una precisión del 99.34 % en la detección de subregiones defectuosas, con un procesamiento rápido en comparación con otras redes CNN.
- La arquitectura propuesta demostró ser particularmente efectiva en textiles con patrones repetitivos o intrincados, donde las técnicas convencionales fallan.

Comentario: Este método es relevante para este caso, ya que las mantas de alpaca presentan patrones complejos que requieren estrategias avanzadas de localización y detección de imperfecciones.

Wei et al. [2021]. *MLMA-Net: Multi-level multi-attentional learning for multi-label object detection in textile defect images*, arXiv preprint arXiv:2102.00376. El estudio fue desarrollado en la Universidad Donghua, ubicada en Shanghái, China, y utilizó el conjunto de datos industrial DHU-ML1000, elaborado por los propios investigadores.

Conclusiones:

- Se desarrolló MLMA-Net, una red neuronal profunda que incorpora módulos multi-nivel y multi-atención para la detección de defectos múltiples y coexistentes en imágenes textiles.
- Utilizando la arquitectura ResNet-50 y módulos especializados, se mejoró notablemente la detección de defectos pequeños y difíciles de identificar.
- El modelo logró altos valores en métricas como precisión, recall y mAP sobre el dataset DHU-ML1000.
- La combinación de aprendizaje multinivel y atención permitió una localización más precisa en entornos industriales con variabilidad en el tipo de defectos.

Comentario: Este trabajo es valioso para la investigación debido a su capacidad de manejar defectos múltiples y de diferentes tamaños, algo común en las mantas artesanales de tejido punto.

Koulali and Eskil [2023]. *Unsupervised textile defect detection using convolutional neural networks*, arXiv preprint arXiv:2312.00224. Como fuente de datos se utilizó el conjunto público *Patterned Fabrics*, compuesto por imágenes de tejidos con patrones. El artículo consultado no especifica una procedencia regional para la captura de todas las imágenes utilizadas.

Conclusiones:

- Se propuso un enfoque no supervisado que combina redes neuronales convolucionales con técnicas de extracción de características y análisis estadístico para detectar imperfecciones sin necesidad de datos etiquetados.
- La metodología utilizó medidas como distancia Manhattan, normalización de imágenes e histogramas para clasificar imperfecciones.
- El modelo logró resultados competitivos frente a métodos supervisados, con menor tiempo y costo computacional.
- Este enfoque reduce significativamente la dependencia de grandes datasets etiquetados, lo que lo hace viable para industrias que no disponen de bases de datos especializadas.

Comentario: Este antecedente es clave para este caso, ya que en la industria textil de alpaca no existe un dataset específico, por lo que un método no supervisado puede acelerar el desarrollo.

Li et al. [2017]. *Deformable patterned fabric defect detection with Fisher criterion-based deep learning*. Investigación desarrollada en la Universidad Donghua, ubicada en Shanghái, China, con imágenes de tejidos estampados deformables.

Conclusiones:

- Se desarrolló un método basado en autoencoders apilados (SDA) optimizados con el criterio de Fisher para mejorar la discriminación de características en tejidos con patrones complejos.
- Las imágenes se dividieron en parches uniformes, clasificándolos como imperfecciones o no imperfecciones mediante análisis de residuo entre imagen reconstruida y parche original.

- El método demostró alta precisión en clasificación y localización de imperfecciones incluso con pocas muestras negativas disponibles.
- La inclusión del criterio de Fisher mejoró la capacidad de detección en escenarios con datos limitados y alta complejidad visual.

Comentario: Este trabajo es útil para el tratamiento de mantas con patrones irregulares o deformables, algo habitual en piezas artesanales, donde las imperfecciones no siguen un patrón fijo.

2.2. Bases Teóricas

2.2.1. Fibra de Alpaca

La fibra de alpaca es un recurso natural altamente valorado en la industria textil. Según la tesis de Azabache-Alvarado et al. [2021], la fibra de alpaca peruana ha experimentado una evolución significativa en términos de producción y exportación entre 2010 y 2019. Durante este período, se observó un crecimiento en la calidad y competitividad de la fibra peruana en el mercado internacional, como se aprecia en la Figura 2.1. Factores como la mejora en las técnicas de manejo y la innovación en el procesamiento han contribuido al posicionamiento de la fibra de alpaca como producto premium.

La fibra de alpaca se distingue por su fineza, suavidad y durabilidad, lo que la hace especialmente apreciada en la confección de textiles de lujo, como mantas de tejido punto. La *baby alpaca* es la más valorada debido a su excepcional suavidad y finura. Además, destaca por su amplia gama de colores naturales, desde tonos claros hasta oscuros, lo que permite prescindir de procesos de teñido en algunos casos. La certificación de origen y sostenibilidad también ha ganado importancia en los últimos años, reforzando su valor en el mercado.

Figura 2.1

Comparación visual de fibra de alpaca.

(a) *Fibra de alpaca negra*



(b) *Fibra de alpaca rubia*



Fuente: Alpaca del Perú [2025].

Propiedades de la Fibra de Alpaca

En un estudio realizado por Cushman [2018], se destacan las propiedades térmicas excepcionales de la fibra de alpaca, que le permiten mantener la temperatura corporal sin causar sobrecalentamiento. Estas características la convierten en una fibra ideal para mantas de tejido punto, ampliamente utilizadas en climas fríos y templados.

Entre sus principales propiedades se destacan:

- **Suavidad:** Mayor que la lana de oveja, lo que aporta confort al usuario.
- **Resistencia:** Alta durabilidad frente al desgaste.
- **Transpirabilidad:** Liviana y con regulación térmica natural.
- **Hipoalergénica:** Libre de lanolina, minimizando reacciones alérgicas.

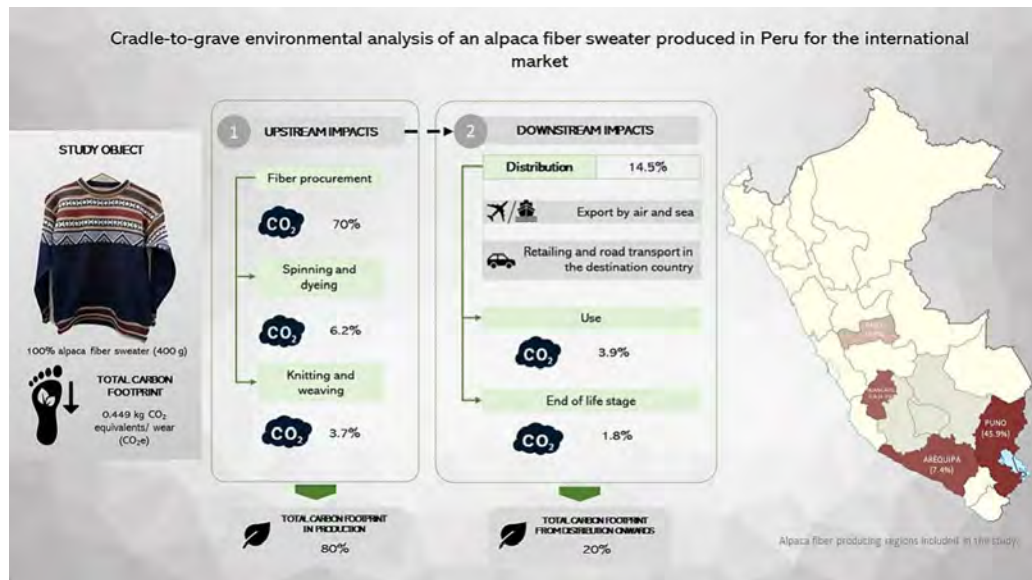
2.2.2. Procesamiento de la Fibra de Alpaca

El procesamiento de la fibra de alpaca es determinante para la calidad de los textiles. Gálvez and Hinojosa [2011] explica que el proceso incluye recolección, lavado, cardado, hilado

y tejido. En mantas de punto, este proceso asegura resistencia, suavidad y uniformidad del material. La Figura 2.2 muestra un análisis del ciclo de vida de la fibra de alpaca.

Figura 2.2

Análisis del ciclo de vida fibra de alpaca.



Fuente: Gálvez and Hinojosa [2011].

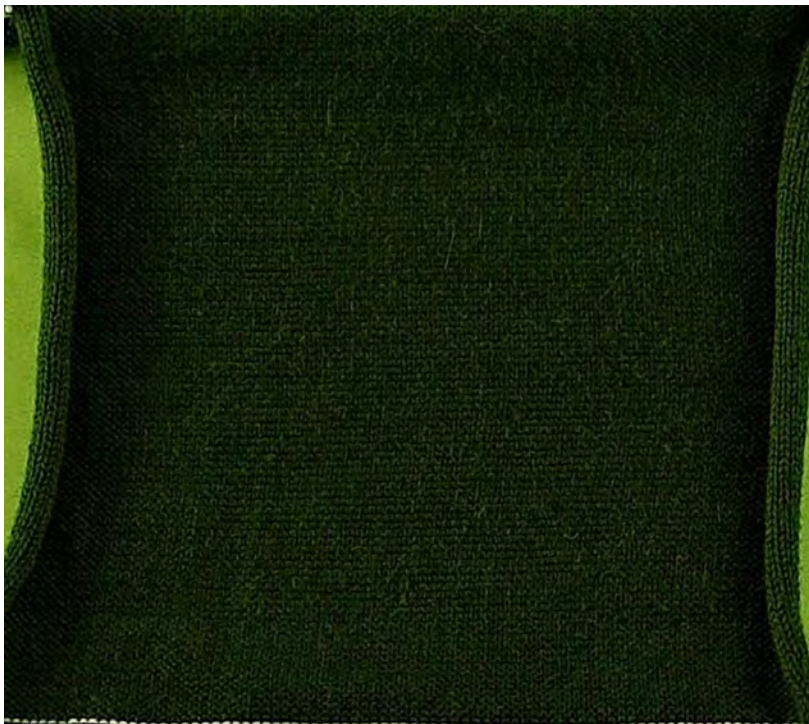
2.2.3. Técnicas de Tejido Punto

El tejido punto, técnica ancestral ampliamente usada en mantas, se caracteriza por su elasticidad y versatilidad. Se basa en entrelazar hilos formando bucles flexibles Hiatt [1989]. Las técnicas más comunes incluyen:

- **Jersey:** Es la estructura más fundamental. Se caracteriza por tener una cara visible con columnas de bucles en forma de "V" y un reverso con una textura ondulada. Su construcción le confiere una notable elasticidad, ligereza y una caída suave, aunque sus bordes tienden a enrollarse naturalmente Black [2005] un ejemplo de ello se ve en la Figura 2.3.

Figura 2.3

Tejido en Punto Jersey.



- **Jacquard:** Se emplea para crear patrones y diseños complejos que son parte integral de la estructura del tejido. Mediante el control selectivo de las agujas, se entrelazan hilos de diferentes colores, lo que resulta en un tejido más grueso, pesado y con menor elasticidad. Es común que en el reverso se observen los hilos no utilizados en la cara, conocidos como flotantes Black [2005], como se observa en la Figura 2.4, siendo muy apreciado en prendas de lujo y textiles decorativos.

Figura 2.4

Tejido en Punto Jacquard.



- **Intarsia:** Es una técnica para tejer bloques de colores sólidos y definidos, similar a un mosaico. Cada área de color se trabaja con un hilo independiente, eliminando los hilos flotantes en el reverso. Esto produce un tejido más ligero que el Jacquard y con una apariencia limpia en ambas caras, ideal para diseños nítidos sin añadir grosor extra a la prenda Black [2005], como se puede observar un ejemplo en la Figura 2.5.

Figura 2.5

Tejido en Punto Intarsia.



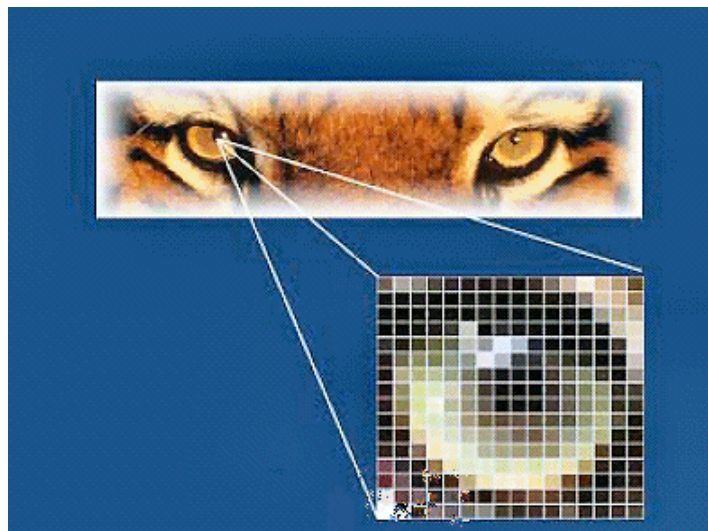
2.2.4. Visión Artificial

La **visión artificial**, también conocida como visión por computadora, es un campo de la inteligencia artificial cuyo objetivo principal es replicar y automatizar las capacidades de la visión humana. No se limita a la simple adquisición de imágenes, sino que abarca un conjunto de procesos complejos para que una máquina pueda interpretar, comprender y tomar decisiones a partir de datos visuales Szeliski [2010]. De acuerdo con Gonzalez y Woods Gonzalez and Woods [2009], la visión artificial busca construir sistemas artificiales que obtienen información útil de las imágenes. El proceso generalmente incluye la adquisición de la imagen, el preprocesamiento, la extracción de características, la segmentación, el reconocimiento de objetos y, finalmente, la interpretación de la escena. En el contexto de la manufactura textil, la implementación de un prototipo de visión artificial permite una inspección de calidad automatizada, objetiva y continua, superando las limitaciones de la inspección humana, como la fatiga, la subjetividad y la velocidad.

2.2.5. Imagen Digital

Una **imagen digital** es la representación numérica de una escena visual bidimensional. Está compuesta por una matriz de elementos discretos llamados píxeles (contracción de *picture elements*), donde cada píxel tiene asociada una ubicación espacial específica y un valor numérico que representa su nivel de intensidad o color Gonzalez and Woods [2009]. La estructura de una imagen digital se define por su resolución espacial (el número de píxeles en la matriz, ej. 1920x1080) y su profundidad de bits o resolución de intensidad (la cantidad de bits utilizados para codificar el valor de cada píxel). Por ejemplo, una imagen en escala de grises de 8 bits puede representar 256 (2^8) niveles diferentes de gris. La calidad y fidelidad de esta representación digital es un factor crítico, ya que toda la información sobre las imperfecciones en el tejido de alpaca debe estar contenida en estos datos numéricos para que los algoritmos posteriores puedan detectarlas. En la siguiente Figura 2.6 se ve la representación de una imagen digital

Figura 2.6
Imagen digital



Fuente: Miguel [2009].

2.2.6. Procesamiento de Imágenes Digitales

El **procesamiento de imágenes digitales** es el conjunto de técnicas y algoritmos que operan sobre una imagen digital con el propósito de limpiar y preparar los datos para su análisis, ya sea para la interpretación humana o para el procesamiento por parte de una máquina. Las operaciones pueden ser de bajo nivel, como la eliminación de ruido o la corrección del contraste, o de alto nivel, como la detección de objetos Bovik [2005]. En la inspección de tejidos, el preprocesamiento es un paso fundamental. Puede incluir técnicas como el filtrado para suavizar la textura natural del tejido sin eliminar los detalles de un defecto, la ecualización del histograma para corregir problemas de iluminación no uniforme, o la transformación a escala de grises para simplificar el análisis. El objetivo es limpiar y estandarizar las imágenes, realzando las características de las imperfecciones para que el modelo de detección sea más efectivo y robusto.

2.2.7. Aprendizaje de Máquina

El **aprendizaje de máquina** (Machine Learning) es una disciplina de la inteligencia artificial que dota a las computadoras de la capacidad de aprender a partir de datos, sin ser explícitamente programadas para una tarea específica Bishop [2006]. En lugar de seguir un conjunto de reglas codificadas, un algoritmo de aprendizaje automático analiza un gran volumen de datos de ejemplo (denominado conjunto de entrenamiento) y aprende a identificar patrones subyacentes. A partir de estos patrones, el modelo resultante puede hacer predicciones o clasificaciones sobre datos nuevos y nunca antes vistos. Este enfoque es particularmente útil para problemas complejos donde la creación de reglas manuales es inviable, como la detección de la sutil y variada gama de imperfecciones en textiles.

2.2.8. Aprendizaje Profundo

El **aprendizaje profundo** (Deep Learning) es un subcampo especializado del aprendizaje de máquina que utiliza redes neuronales con una arquitectura "profunda", es decir, con múltiples capas ocultas entre la capa de entrada y la de salida. Esta profundidad permite al modelo aprender una jerarquía de características, donde las capas iniciales aprenden características simples (como bordes o texturas) y las capas más profundas combinan estas para aprender conceptos más complejos y abstractos (como partes de un defecto o el defecto en su totalidad) Goodfellow et al. [2016]. Esta capacidad de aprendizaje automático de representaciones jerárquicas es lo que ha llevado al éxito sin precedentes del aprendizaje profundo en tareas de visión por computadora, superando a los métodos tradicionales que requerían una ingeniería de características manual y experta.

2.2.9. Redes Neuronales Artificiales

Las **Redes Neuronales Artificiales (RNA)** son modelos matemáticos inspirados en la estructura y el funcionamiento de las redes neuronales biológicas del cerebro. Están compuestas por un conjunto de nodos interconectados, o "neuronas", organizados en capas. Cada conexión entre neuronas tiene un peso sináptico asociado, que se ajusta durante el proceso de entrenamiento mediante un algoritmo de optimización. La red aprende a mapear entradas a salidas procesando la información a través de sus capas Haykin [2009]. Las RNA son sistemas adaptativos capaces de modelar relaciones extremadamente complejas y no lineales, lo que las hace idóneas para tareas de clasificación y reconocimiento de patrones.

2.2.10. Redes Neuronales Convolucionales

Las **Redes Neuronales Convolucionales (CNN)** son una clase de redes neuronales profundas que ha demostrado ser excepcionalmente efectiva en el análisis de datos

visuales. Su diseño está inspirado en el córtex visual de los animales. La arquitectura de una CNN se diferencia de una red neuronal convencional por el uso de capas convolucionales, que aplican filtros para detectar patrones locales, y capas de agrupamiento (pooling), que reducen la dimensionalidad de los datos. Esta estructura les permite tener invarianza a la traslación y aprender jerarquías de características espaciales de manera eficiente, convirtiéndolas en el estándar de oro para tareas como la clasificación de imágenes y la detección de objetos LeCun et al. [2015].

Capas de las Redes Neuronales Convolucionales

La arquitectura de una CNN se compone de una secuencia de capas, cada una con una función específica.

Capa de Entrada. Es la primera capa de la red y su función es recibir los datos brutos. Para un problema de visión por computadora, esta capa toma la forma de una matriz de píxeles que representa la imagen. Sus dimensiones son alto $\times$ ancho $\times$ canales de color (ej. 1 para escala de grises, 3 para RGB) Goodfellow et al. [2016].

Capa de Convolución. Es el bloque de construcción fundamental de una CNN. En esta capa, un conjunto de filtros (o kernels), que son pequeñas matrices de pesos, se deslizan a través de la imagen de entrada realizando una operación de convolución. Esta operación calcula un producto punto entre el filtro y una porción de la imagen, produciendo un mapa de características. Cada filtro se especializa en detectar un patrón específico, como un borde vertical, una esquina o una textura particular Chollet [2017].

Capa de Agrupamiento (Pooling). Esta capa, comúnmente insertada entre capas de convolución, se encarga de reducir progresivamente el tamaño espacial de la representación para disminuir la cantidad de parámetros y la carga computacional. Además, confiere al

modelo una mayor robustez frente a pequeñas variaciones en la posición de las características en la imagen. La operación más común es *Max Pooling*, que extrae el valor máximo de una ventana del mapa de características, conservando así la información más prominente Goodfellow et al. [2016].

2.2.11. Optimizadores en Redes Neuronales Profundas

El entrenamiento de una red neuronal es un problema de optimización en el que se busca encontrar el conjunto de pesos que minimice una función de pérdida (la medida del error del modelo). Los **optimizadores** son los algoritmos que implementan la estrategia para actualizar estos pesos de manera iterativa, basándose en el gradiente de la función de pérdida.

Adam (Adaptive Moment Estimation) es uno de los algoritmos de optimización más populares y efectivos. Su principal ventaja es que adapta la tasa de aprendizaje para cada peso de la red de forma individual. Lo hace utilizando estimaciones de los momentos de primer y segundo orden de los gradientes, es decir, mantiene un promedio móvil tanto de los gradientes como del cuadrado de los gradientes. Esto le permite converger más rápidamente y de manera más estable que otros optimizadores estocásticos Kingma and Ba [2014].

2.2.12. Métricas para la evaluación de modelos de detección

La evaluación de un modelo de detección de objetos requiere considerar tanto la clasificación de cada objeto como la localización del cuadro delimitador. Para determinar si una predicción es correcta, se compara el cuadro predicho con la anotación real mediante la métrica de intersección sobre unión (IoU) y se fija un umbral mínimo de coincidencia. A partir de esta comparación se definen los verdaderos positivos (TP), falsos positivos (FP) y falsos negativos (FN).

Un verdadero positivo corresponde a una imperfección detectada con la clase correcta y con una superposición igual o superior al umbral de IoU establecido. Un falso positivo es una detección que no corresponde a una imperfección real, que presenta una clase incorrecta o que no alcanza el umbral de IoU. Un falso negativo es una imperfección presente en la imagen que el modelo no detecta Fawcett [2006].

Precisión

La precisión expresa qué proporción de las detecciones realizadas por el modelo es correcta. Se calcula mediante:

$$\text{Precisión} = \frac{TP}{TP + FP}. \quad (2.1)$$

Un valor elevado indica que el modelo genera pocas falsas alarmas. En esta investigación, permite conocer qué tan confiables son las detecciones señaladas como huecos o puntos corridos.

Recall o sensibilidad

El *recall* mide la proporción de imperfecciones reales que fueron detectadas por el modelo:

$$\text{Recall} = \frac{TP}{TP + FN}. \quad (2.2)$$

Un valor elevado indica que pocas imperfecciones quedaron sin detectar. Esta métrica es especialmente relevante en la inspección textil, ya que un falso negativo representa un hueco o punto corrido que no fue advertido por el modelo.

Intersección sobre unión

La intersección sobre unión (IoU) cuantifica el grado de superposición entre el cuadro delimitador predicho, B_p , y el cuadro real, B_r :

$$\text{IoU} = \frac{|B_p \cap B_r|}{|B_p \cup B_r|}. \quad (2.3)$$

El valor de IoU varía entre 0 y 1. Un valor de 0 indica ausencia de superposición, mientras que un valor de 1 representa coincidencia perfecta. Por tanto, IoU evalúa qué tan correctamente fue localizada la imperfección, además de que esta haya sido clasificada.

Precisión promedio y mAP@50

La precisión promedio (AP) de una clase corresponde al área bajo la curva precisión–recall obtenida al variar el umbral de confianza. La precisión promedio media (mAP) se obtiene promediando la AP de las C clases:

$$\text{AP}_c = \int_0^1 P_c(R) dR, \quad (2.4)$$

$$\text{mAP} = \frac{1}{C} \sum_{c=1}^C \text{AP}_c. \quad (2.5)$$

En la métrica mAP@50, una detección puede considerarse correcta cuando alcanza un IoU mínimo de 0.50. En esta investigación, el resultado corresponde al promedio de las clases hueco y punto corrido. Debido a que utiliza un único umbral de superposición relativamente flexible, mAP@50 refleja principalmente la capacidad general de detectar y localizar las imperfecciones.

mAP@50–95

La métrica mAP@50–95 es más estricta, pues promedia el mAP calculado en diez umbrales de IoU, desde 0.50 hasta 0.95 con incrementos de 0.05 Lin et al. [2014]:

$$\text{mAP@50–95} = \frac{1}{10} \sum_{t \in \{0.50, 0.55, \dots, 0.95\}} \text{mAP}_t. \quad (2.6)$$

Al incluir umbrales elevados, esta métrica penaliza con mayor intensidad los cuadros delimitadores imprecisos. En consecuencia, ofrece una valoración más completa de la clasificación y de la calidad de localización.

F1-score

El F1-score corresponde a la media armónica entre precisión y *recall*:

$$F1 = 2 \frac{\text{Precisión} \cdot \text{Recall}}{\text{Precisión} + \text{Recall}}. \quad (2.7)$$

Esta métrica resume el equilibrio entre evitar falsas alarmas y detectar la mayor cantidad posible de imperfecciones. Un F1-score alto requiere que tanto la precisión como el *recall* presenten valores elevados.

2.2.13. Técnicas de Evaluación de Modelos

Validación Cruzada Es una técnica estadística para evaluar la capacidad de generalización de un modelo. En lugar de una única división de datos en entrenamiento y prueba, la técnica *k-fold cross-validation* divide el conjunto de datos en k subconjuntos o "pliegues". El modelo se entrena y evalúa k veces, utilizando en cada iteración un pliegue diferente para la prueba y los $k - 1$ restantes para el entrenamiento. El rendimiento final se promedia sobre las k iteraciones, proporcionando una estimación mucho más robusta y menos sesgada del

rendimiento del modelo en datos no vistos Kohavi [1995].

Matriz de Confusión Es una tabla que resume el rendimiento de un modelo de clasificación. Las filas representan las clases predichas por el modelo y las columnas representan las clases reales. La diagonal principal muestra las predicciones correctas, mientras que los elementos fuera de la diagonal revelan los errores de clasificación y, más importante aún, qué clases se están confundiendo entre sí. Es una herramienta indispensable para un análisis de error detallado Fawcett [2006].

Curva ROC Es una herramienta gráfica para visualizar el rendimiento de un clasificador binario. Grafica la Tasa de Verdaderos Positivos (Sensibilidad) en el eje Y frente a la Tasa de Falsos Positivos (1 - Especificidad) en el eje X para diferentes umbrales de decisión. Un modelo perfecto tendría una curva que pasa por la esquina superior izquierda. El **Área Bajo la Curva (AUC)** es una métrica que resume el rendimiento del clasificador en una sola cifra, donde 1.0 representa un clasificador perfecto Fawcett [2006].

2.2.14. Transferencia de Aprendizaje

La **transferencia de aprendizaje** (Transfer Learning) es una metodología de aprendizaje automático que ha revolucionado la aplicación práctica del aprendizaje profundo. Consiste en tomar un modelo previamente entrenado en una tarea a gran escala (por ejemplo, una CNN entrenada para clasificar millones de imágenes del dataset ImageNet) y reutilizarlo como punto de partida para una tarea diferente pero relacionada (como la detección de imperfecciones en tejidos). Las capas iniciales de estos modelos pre-entrenados han aprendido a detectar características visuales genéricas (bordes, colores, texturas) que son útiles para casi cualquier tarea de visión. Al aprovechar este conocimiento, se puede entrenar un modelo de alto rendimiento con un conjunto de datos mucho más pequeño y en menos tiempo Pan

and Yang [2009].

2.2.15. Data Augmentation

El **aumento de datos** (Data Augmentation) es un conjunto de técnicas para incrementar artificialmente el tamaño y la diversidad del conjunto de datos de entrenamiento. Esto se logra generando nuevas muestras de datos a partir de las existentes mediante la aplicación de transformaciones que no alteran la semántica de la etiqueta. El objetivo es exponer al modelo a una mayor variabilidad de condiciones, lo que mejora su capacidad de generalización y reduce el sobreajuste (overfitting), que ocurre cuando un modelo aprende demasiado bien los datos de entrenamiento pero funciona mal con datos nuevos Shorten and Khoshgoftaar [2019].

Técnicas de Aumento de Datos en Imágenes

En el dominio de las imágenes, las técnicas de aumento de datos son particularmente efectivas. Algunas de las transformaciones más comunes incluyen:

- **Volteo:** El volteo de imágenes consiste en invertirlas de manera horizontal o vertical. Esta técnica permite duplicar o incluso cuadruplicar el tamaño del conjunto de datos original, alcanzando un factor de aumento aproximado de 2x a 4x. El volteo es especialmente útil para que el modelo aprenda a identificar imperfecciones textiles sin depender de una orientación específica del tejido.
- **Rotación:** La rotación implica girar la imagen en diferentes ángulos, ya sea en múltiplos de 90 grados o en ángulos arbitrarios. Un aspecto importante de esta técnica es que las dimensiones de la imagen pueden no conservarse después de la rotación, especialmente cuando se trata de imágenes rectangulares o rotaciones con ángulos distintos a 90 o 180 grados. En estos casos, es necesario aplicar técnicas de redimensionamiento

Figura 2.7

Aumento de datos mediante volteo

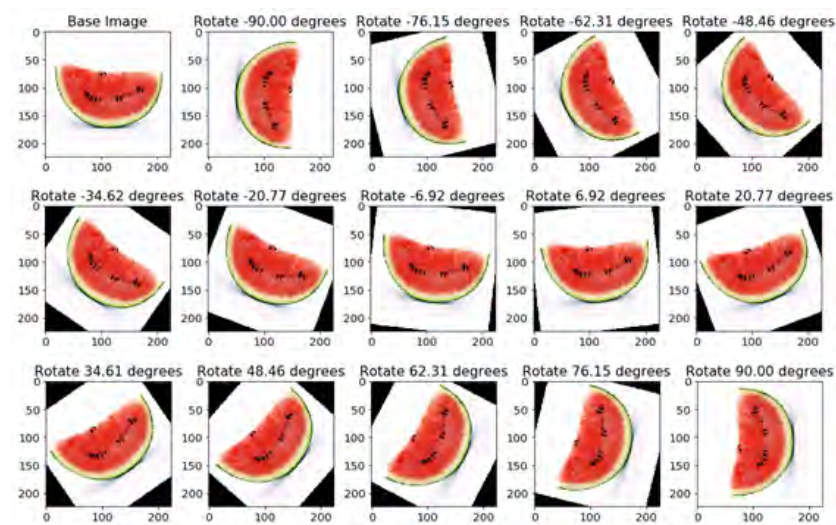


Fuente. Calvo [2024]

o recorte para mantener un tamaño uniforme. La rotación permite simular variaciones en la posición del tejido durante la inspección, con un factor de aumento de datos aproximado de 2x a 4x.

Figura 2.8

Aumento de datos mediante Rotacion



Fuente. Calvo [2024]

- **Escalado:** El escalado consiste en modificar el tamaño de la imagen, ya sea ampliándola o reduciéndola. Al escalar hacia afuera, la imagen resultante es mayor que la original, por lo que se recorta una región del mismo tamaño que la imagen inicial. En el escalado hacia adentro, la imagen se reduce, lo que implica asumir información fuera de los límites originales. Esta técnica permite que el modelo aprenda a detectar imperfecciones

de diferentes tamaños y proporciones. El factor de crecimiento del conjunto de datos mediante escalado es arbitrario, dependiendo de los parámetros utilizados.

Figura 2.9

aumento de datos mediante Escalado



Fuente. Calvo [2024]

- **Recorte:** El recorte consiste en seleccionar de manera aleatoria una subregión de la imagen original y posteriormente redimensionarla para que coincida con el tamaño inicial. Este método, conocido como recorte aleatorio, no modifica toda la imagen, sino únicamente una sección específica. El recorte favorece el aprendizaje de características locales y es especialmente útil para resaltar imperfecciones puntuales como huecos o puntos corridos. Su factor de aumento de datos también es arbitrario.
- **Traslación:** La traslación implica desplazar la imagen en los ejes horizontal, vertical o ambos. Durante este proceso, se asume que el área fuera de los límites originales de la imagen es rellenada con un fondo uniforme. Esta técnica es altamente efectiva, ya que las imperfecciones textiles pueden aparecer en cualquier posición dentro de la manta. Al aplicar traslaciones, el modelo se ve obligado a buscar imperfecciones en distintas ubicaciones, mejorando su capacidad de generalización. El factor de crecimiento de datos generado mediante traslación es arbitrario.

Figura 2.10

aumento de datos mediante Recorte



Fuente. Calvo [2024]

Figura 2.11

aumento de datos mediante Traslación



Fuente. Calvo [2024]

- **Cizallamiento:** El cizallamiento (*shear*) es una transformación geométrica que inclina la imagen horizontal o verticalmente, desplazando progresivamente sus puntos sin introducir una deformación de perspectiva. Esta técnica permite representar pequeñas variaciones angulares ocasionadas por la posición de la cámara o de la superficie capturada. En YOLOv5, el cizallamiento forma parte de las transformaciones geométricas aleatorias aplicadas durante el entrenamiento Ultralytics [2026].

La aplicación de una combinación aleatoria de estas transformaciones a cada imagen durante el entrenamiento obliga al modelo a aprender las características esenciales del objeto de interés (el defecto), independientemente de su orientación, posición o condiciones de iluminación Shorten and Khoshgoftaar [2019].

2.2.16. Arquitecturas de Redes Neuronales Convolucionales

El progreso en la visión por computadora ha estado marcado por el desarrollo de arquitecturas de CNN cada vez más sofisticadas. Arquitecturas como **AlexNet**, que popularizó las CNN; **VGGNet**, que demostró la importancia de la profundidad; **GoogLeNet (Inception)**, que introdujo módulos computacionalmente eficientes; y **ResNet**, que permitió entrenar redes extremadamente profundas mediante conexiones residuales, han sido hitos fundamentales. Cada una de estas arquitecturas ha contribuido con conceptos clave que han impulsado el estado del arte Khan et al. [2020].

2.2.17. YOLO

YOLO (You Only Look Once) es una familia de modelos de detección de objetos que se destaca por su enfoque innovador de una sola etapa. A diferencia de los detectores de dos etapas que primero proponen regiones de interés y luego las clasifican, YOLO trata la detección de objetos como un único problema de regresión. Divide la imagen en una cuadrícula y, para cada celda, predice simultáneamente los cuadros delimitadores (bounding boxes) y las probabilidades de clase. Este diseño unificado permite una velocidad de inferencia extremadamente alta, haciéndolo ideal para aplicaciones en tiempo real Redmon et al. [2016]. **YOLOv5**, una de las versiones más recientes, mantenida por Ultralytics, no solo refina esta arquitectura para lograr un excelente equilibrio entre velocidad y precisión, sino que también incorpora las mejores prácticas de ingeniería de software, como el aumento de datos integrado y la facilidad de entrenamiento y despliegue Ultralytics [2020].

Capítulo 3

Desarrollo del tema de tesis

3.1. Metodología

La metodología de investigación adoptada en el presente trabajo se basa en el enfoque cuantitativo propuesto por Hernández-Sampieri et al. [2014], dado que se busca medir el desempeño, precisión y eficacia de modelos de visión computacional en la detección de imperfecciones en mantas de tejido punto elaboradas con fibras de alpaca.

El tipo de investigación es aplicada, ya que se orienta al desarrollo e implementación de una solución tecnológica destinada a mejorar el proceso de control de calidad en el sector textil. Asimismo, el nivel de investigación es descriptivo, puesto que se analizan las relaciones entre las características visuales del tejido y la presencia de imperfecciones de manufactura.

Para el desarrollo del prototipo de detección de imperfecciones se siguieron las siguientes etapas:

1. Revisión bibliográfica.
2. Recolección de imágenes de mantas de tejido punto de alpaca.
3. Preprocesamiento de imágenes.

4. Entrenamiento de modelos de visión computacional.
5. Integración del modelo en un en un prototipo de detección de imperfecciones.

3.2. Recolección y construcción del conjunto de datos

La recolección y construcción del conjunto de datos constituyó una etapa fundamental para el desarrollo del prototipo de detección de imperfecciones. En esta fase se capturaron imágenes de mantas de tejido punto elaboradas con fibras de alpaca, considerando muestras con imperfecciones de manufactura y muestras sin imperfecciones visibles.

3.2.1. Definición de clases

A partir de la inspección del proceso productivo se definieron dos clases de imperfecciones:

- **Huecos (*gap*)**: imperfecciones caracterizadas por la ausencia parcial o total de puntos en la estructura del tejido, lo que genera aberturas visibles que afectan la resistencia y la apariencia del producto.
- **Puntos corridos (*running-stitch*)**: imperfecciones producidas cuando una secuencia de puntos se libera o se desplaza de su posición original, generando una deformación lineal visible en el tejido.

Las muestras sin imperfecciones no fueron consideradas como una tercera clase, debido a que no representan un objeto que el modelo deba localizar. Estas imágenes se incorporaron como muestras de fondo para ayudar al modelo a distinguir entre una imperfección real y la textura normal del tejido.

3.2.2. Proceso de captura de imágenes

Las imágenes fueron capturadas con un dispositivo móvil Xiaomi Redmi Note 9, equipado con una cámara de 48 MP. La captura se realizó a una distancia aproximada de 1,5 metros, colocando las mantas sobre una superficie plana con un fondo uniforme de color verde, con el propósito de reducir las interferencias visuales del entorno.

Se consideraron 14 diseños diferentes de mantas de tejido punto de alpaca, lo que permitió incorporar variaciones de textura, patrón y color. Durante la captura se empleó iluminación natural controlada y se procuró evitar sombras pronunciadas y reflejos sobre el tejido.

3.2.3. Anotación y composición del conjunto de datos

El conjunto de datos quedó conformado por 549 imágenes. De este total, 449 correspondieron a mantas con huecos o puntos corridos y fueron anotadas manualmente mediante cajas delimitadoras. Las 100 imágenes restantes correspondieron a mantas sin imperfecciones visibles y se utilizaron como muestras de fondo.

Las anotaciones se almacenaron en archivos de texto siguiendo el formato requerido por YOLO. Para cada objeto se registró la clase y las coordenadas normalizadas de su caja delimitadora. En el caso de las imágenes sin imperfecciones, se utilizaron archivos de anotación vacíos, debido a que estas muestras no contenían objetos pertenecientes a las clases definidas.

Por consiguiente, el modelo se configuró únicamente con las siguientes clases:

- Clase 0: *gap* (huecos).
- Clase 1: *running-stitch* (puntos corridos).

3.2.4. División y organización del conjunto de datos

El conjunto de datos se dividió en los subconjuntos de entrenamiento, validación y prueba, utilizando una proporción de 70 %, 15 % y 15 %, respectivamente. La distribución obtenida se presenta en la Tabla 3.1.

Tabla 3.1

Distribución del conjunto de datos

Conjunto	Cantidad de imágenes	Porcentaje aproximado
Entrenamiento	385	70 %
Validación	82	15 %
Prueba	82	15 %
Total	549	100 %

Durante la división se procuró mantener una representación adecuada de las clases y de las imágenes sin imperfecciones en los tres subconjuntos. El subconjunto de entrenamiento se utilizó para ajustar los parámetros del modelo; el de validación, para supervisar su desempeño durante el entrenamiento; y el de prueba, para realizar la evaluación final.

El conjunto de datos se organizó mediante directorios separados para las imágenes y sus respectivas etiquetas:

- `train/`: imágenes y etiquetas utilizadas durante el entrenamiento.
- `val/`: imágenes y etiquetas utilizadas durante la validación.
- `test/`: imágenes y etiquetas reservadas para la evaluación final.

Las rutas y las clases del conjunto de datos se declararon en el archivo `config.yaml`, cuya configuración fue la siguiente:

```
1 path: /content/data
2 train: images/train
3 val: images/val
4 test: images/test
5
```

```
6 names :  
7   0: gap  
8   1: running-stitch
```

Código 3.1: Configuración del conjunto de datos en formato YOLO

3.2.5. Publicación y disponibilidad del conjunto de datos

El conjunto de datos fue publicado en Kaggle para facilitar su consulta y reutilización. Contiene 549 imágenes de mantas de tejido de alpaca, anotadas en formato YOLO para las clases *hueco* y *punto corrido*, y conserva la división de entrenamiento, validación y prueba.

Disponible en: <https://www.kaggle.com/datasets/estefans008/alpaca-knitted-blanket-defect-dataset>.

3.3. Preprocesamiento de imágenes

El preprocesamiento de imágenes comprende un conjunto de operaciones destinadas a limpiar y preparar las imágenes digitales antes de su análisis por parte de un modelo de visión computacional. Estas operaciones permiten reducir elementos no deseados, corregir variaciones visuales y estandarizar los datos de entrada, procurando conservar las características relevantes de los objetos que se desean detectar Bovik [2005].

Durante la inspección del conjunto de datos se identificaron variaciones de iluminación, arrugas en el tejido y presencia de ruido visual. Estos elementos podían dificultar la identificación de los huecos y puntos corridos, especialmente debido al reducido tamaño de estas imperfecciones. Por esta razón, se evaluaron diferentes técnicas de preprocesamiento orientadas a atenuar el ruido y uniformizar las condiciones visuales de las imágenes. La aplicación de estas técnicas se realizó procurando no eliminar ni alterar las características de las imperfecciones presentes en el tejido, debido a que estas constituían la información principal para el entrenamiento del modelo.

3.3.1. Evaluación y selección de filtros de preprocesamiento

Antes del entrenamiento se analizaron diferentes filtros de procesamiento de imagen con el propósito de determinar si contribuían a reducir el ruido sin eliminar las características visuales de las imperfecciones. La evaluación fue cualitativa y se realizó observando una misma muestra de imágenes con huecos, puntos corridos y tejido sin imperfecciones. Se examinó principalmente la conservación de los bordes, la textura del tejido y la visibilidad de las imperfecciones pequeñas. Esta evaluación no se utilizó para comparar los modelos finales, sino para seleccionar un preprocesamiento que no modificara de manera perjudicial la información de entrada.

Los filtros de suavizado gaussiano, mediana y bilateral poseen efectos distintos sobre el ruido y los bordes OpenCV [2024]; por su parte, los filtros de Gabor permiten analizar frecuencia y orientación y han sido utilizados en la detección de anomalías textiles Jing et al. [2011]. No obstante, las mantas analizadas presentan textura irregular, fibras sobresalientes, diferentes colores y detalles de pequeña escala. Un preprocesamiento intenso podía suprimir o alterar precisamente las señales que el modelo debía aprender. Por ello, se conservó mayormente la información RGB original y el filtro gaussiano se aplicó solo de manera ligera en las imágenes que presentaban ruido visible. Los demás filtros no se incorporaron al flujo definitivo, al no observarse una mejora visual consistente para los distintos tipos de manta.

3.3.2. Evaluación de transformaciones para el aumento de datos

Las transformaciones geométricas se analizaron considerando la configuración de captura: la cámara se ubicó aproximadamente perpendicular a la superficie y la manta permaneció extendida. En YOLOv5, la rotación, la traslación, el escalado y el cizallamiento se aplican conjuntamente mediante una transformación afín aleatoria; los volteos horizontal y vertical se controlan de forma independiente Ultralytics [2026]. Por tanto, no resulta correcto

Tabla 3.2*Filtros analizados durante el preprocesamiento de imágenes*

Filtro	Finalidad	Criterio de selección o descarte
Gaussiano	Atenuar ruido y variaciones locales suaves.	Se aplicó de manera limitada y con un kernel reducido, debido a que un suavizado intenso podía difuminar los bordes de huecos pequeños y puntos corridos.
Mediana	Reducir ruido impulsivo conservando discontinuidades.	No se incorporó al flujo final porque podía sustituir píxeles asociados a detalles finos de las fibras y modificar imperfecciones de poca extensión.
Bilateral	Suavizar regiones y conservar bordes.	Aunque mantuvo mejor los contornos que el filtro gaussiano, presentó mayor costo computacional y podía homogeneizar variaciones locales útiles para distinguir la imperfección de la textura.
Gabor	Resaltar patrones de frecuencia y orientación de la textura textil.	Fue considerado por su aplicación en análisis de tejidos; sin embargo, la variedad de puntos, colores y orientaciones exigía ajustar múltiples frecuencias y ángulos. Además, podía intensificar el patrón normal del tejido y generar respuestas que no correspondían a imperfecciones.
CLAHE	Incrementar el contraste local.	No se utilizó como transformación fija porque podía amplificar simultáneamente fibras, sombras y variaciones del tejido, creando contrastes que no estaban presentes en la captura original.
Sobel/Canny	Resaltar bordes y cambios bruscos de intensidad.	No se incorporó porque producía numerosos contornos asociados al tejido normal y podía fragmentar la representación visual de los huecos y puntos corridos.

indicar que una sola transformación engloba también ambos volteos.

Tabla 3.3

Transformaciones analizadas para el aumento de datos

Transformación	Configuración	Justificación
Transformación afín aleatoria	Rotación $\pm 10^\circ$, traslación $\pm 20\%$, escala variable y cizallamiento $\pm 2^\circ$	Se utilizó para representar pequeñas variaciones de posicionamiento y distancia sin deformar excesivamente la geometría de las imperfecciones.
Volteo horizontal	Probabilidad 0.50	Se utilizó porque la ubicación izquierda-derecha no modifica la clase de una imperfección y permite ampliar la variabilidad espacial.
Volteo vertical	Probabilidad 0.00	No se utilizó para conservar la orientación longitudinal del tejido y del punto corrido respecto del proceso de captura.
Perspectiva	Valor 0.00	No se utilizó porque la cámara se dispuso de forma perpendicular a la manta; una deformación proyectiva intensa no representaría la geometría prevista del sistema.
Rotaciones amplias	No utilizadas	No se aplicaron giros de 90° o superiores, pues se apartaban de la orientación de captura y podían introducir relleno o recorte excesivo en los bordes.
Deformación elástica	No utilizada	Se descartó porque modifica localmente la geometría del tejido y podía alterar artificialmente la forma de los huecos o puntos corridos.
Mosaic	Probabilidad 1.00	Se utilizó para combinar cuatro imágenes de entrenamiento, incrementar la diversidad de fondos, escalas y posiciones, y exponer al modelo a varias imperfecciones en una misma composición.
MixUp	Probabilidad 0.15	Se utilizó con baja probabilidad para combinar información de dos imágenes sin dominar la apariencia original del tejido.
Ajustes HSV	$h = 0,015$, $s = 0,7$, $v = 0,4$	Se utilizaron para simular variaciones moderadas de tono, saturación e iluminación.

La selección evitó transformaciones que produjeran escenarios incompatibles con la captura cenital o que modificaran de forma artificial las características morfológicas de las imperfecciones. Todas las transformaciones se aplicaron únicamente al conjunto de entrenamiento; los conjuntos de validación y prueba mantuvieron las imágenes sin aumento artificial para proporcionar una evaluación independiente.

Aunque el archivo de hiperparámetros incluyó `copy_paste: 0.1`, el conjunto de datos

fue anotado mediante cuadros delimitadores y no mediante máscaras de segmentación. Por ello, esta opción no debe presentarse como una transformación efectiva del experimento sin comprobar su ejecución en la versión exacta de YOLOv5 utilizada.

3.4. Justificación de la arquitectura YOLOv5L

Se seleccionó la familia YOLO por tratarse de detectores de una etapa capaces de realizar la localización y clasificación de múltiples objetos en una sola pasada, característica adecuada para una posterior ejecución sobre video. En particular, YOLOv5 ofrece una implementación en PyTorch, pesos preentrenados, entrenamiento por transferencia y herramientas integradas para aumento de datos, validación e inferencia. Su arquitectura combina extracción jerárquica de características y fusión multiescala, lo cual resulta pertinente cuando los objetos de interés ocupan regiones pequeñas dentro de una imagen de alta resolución Ultralytics [2026].

Dentro de la familia YOLOv5 se analizaron inicialmente las variantes S, M y L. Las variantes de menor tamaño requieren menos recursos, pero poseen menor capacidad para representar patrones visuales finos. Se seleccionó YOLOv5L debido a su mayor capacidad de extracción de características y al mejor comportamiento observado en las pruebas preliminares para localizar huecos y puntos corridos de pequeña escala. La resolución de entrada de 1024×1024 píxeles permitió conservar mayor detalle espacial que una resolución inferior, mientras que el tamaño de lote de 8 imágenes se ajustó a la memoria disponible durante el entrenamiento.

3.5. Entrenamiento de modelos

Con el objetivo de evaluar el desempeño de YOLOv5L en la detección automática de imperfecciones en mantas de tejido punto de alpaca, se realizaron cuatro experimentos comparativos. En todos los experimentos se utilizó el conjunto de datos descrito anteriormente, manteniendo la misma división de entrenamiento, validación y prueba para garantizar una comparación bajo condiciones equivalentes.

Los experimentos se diferenciaron por la configuración de la función de pérdida, las técnicas de aumento de datos y el optimizador utilizado. En todos los casos se empleó una resolución de entrada de 1024×1024 píxeles, un tamaño de lote de 8 imágenes y un máximo de 100 épocas. Asimismo, se aplicó parada temprana (*early stopping*) con una paciencia de 20 épocas.

El Experimento 1 se utilizó como línea base; el Experimento 2 incorporó Focal Loss; el Experimento 3 aplicó un aumento de datos avanzado; y el Experimento 4 empleó AdamW junto con una programación cosenoidal de la tasa de aprendizaje.

3.5.1. Definición de la arquitectura YOLOv5L

La arquitectura seleccionada fue YOLOv5L, una variante de alta capacidad dentro de la familia YOLOv5. Su elección respondió a la necesidad de extraer características visuales suficientemente detalladas para localizar imperfecciones textiles de pequeña escala y con formas irregulares dentro de imágenes de alta resolución.

YOLOv5L está compuesto por tres elementos principales: un extractor de características o *backbone*, un cuello de fusión multiescala o *neck* y una cabeza de detección. El *backbone*, basado en estructuras CSP, obtiene representaciones visuales con diferentes niveles de profundidad. Posteriormente, el *neck* combina características de distintas escalas para conservar información global y localizada. Finalmente, la cabeza de detección predice

la clase, la probabilidad de presencia del objeto y las coordenadas de la caja delimitadora.

La capacidad de generar predicciones en diferentes escalas resultó pertinente para este estudio, debido a que los huecos y los puntos corridos podían ocupar regiones reducidas dentro de la imagen. Asimismo, YOLOv5L permitió mantener un equilibrio entre capacidad de representación, precisión de detección y tiempo de inferencia.

3.5.2. Diseño experimental

Los cuatro experimentos fueron diseñados para evaluar el efecto de una estrategia determinada. Los parámetros generales se conservaron sin cambios, excepto cuando la naturaleza del experimento exigió modificar el optimizador, la tasa de aprendizaje, la regularización o las transformaciones de aumento de datos. La Tabla 3.4 resume las configuraciones evaluadas.

Tabla 3.4

Configuraciones evaluadas durante el entrenamiento

Exp.	Modelo	Estrategia evaluada	Optimizador
1	YOLOv5L	Configuración base con los aumentos estándar definidos para el entrenamiento	SGD
2	YOLOv5L	Incorporación de Focal Loss con $\gamma = 2,0$	SGD
3	YOLOv5L	Aplicación de una configuración de aumento de datos avanzado	SGD
4	YOLOv5L	AdamW con programación cosenoidal de la tasa de aprendizaje	AdamW

3.5.3. Entorno y procedimiento de entrenamiento

Los experimentos se ejecutaron en Google Colab utilizando los recursos de procesamiento gráfico proporcionados por la plataforma. El conjunto de datos y el archivo `config.yaml` fueron almacenados en Google Drive y posteriormente copiados al entorno de ejecución. A continuación, se descomprimió el conjunto de datos, se clonó el repositorio de

YOLOv5 y se instalaron las dependencias requeridas.

En todos los experimentos se emplearon los pesos preentrenados `yolov5l.pt`. Este procedimiento permitió aplicar aprendizaje por transferencia, utilizando como punto de partida características visuales aprendidas previamente y ajustándolas al dominio de los defectos textiles estudiados.

Las imágenes fueron procesadas con una resolución de entrada de 1024×1024 píxeles y se utilizó un tamaño de lote de 8 imágenes. Cada configuración fue entrenada durante un máximo de 100 épocas y se estableció una paciencia de 20 épocas para la parada temprana (*early stopping*). Por consiguiente, el entrenamiento finalizaba cuando la métrica de aptitud utilizada internamente por YOLOv5 no presentaba mejoras durante 20 épocas consecutivas.

Durante cada ejecución se generaron los archivos `last.pt`, correspondiente al estado de la última época, y `best.pt`, correspondiente a los pesos de la época con mejor aptitud en el conjunto de validación. Para las evaluaciones posteriores se utilizó el archivo `best.pt` de cada experimento.

Con fines de reproducibilidad, en el cuaderno de Google Colab se incluyeron los comandos `git rev-parse HEAD` y `nvidia-smi`, destinados a registrar el identificador exacto del *commit* de YOLOv5 y el modelo de GPU asignado. El código completo del Experimento 3 se presenta en el Anexo A.

3.5.4. Configuración común de entrenamiento

La Tabla 3.5 presenta los parámetros generales y los hiperparámetros que se mantuvieron constantes en los cuatro experimentos.

Tabla 3.5*Configuración común utilizada en los cuatro experimentos*

Parámetro	Valor
Arquitectura	YOLOv5L
Pesos iniciales	yolov5l.pt
Resolución de entrada	1024 × 1024 píxeles
Tamaño de lote	8 imágenes
Número máximo de épocas	100
Paciencia de parada temprana	20 épocas
Momento inicial de calentamiento (warmup_momentum)	0.8
Tasa para el sesgo durante el calentamiento (warmup_bias_lr)	0.1
Ganancia de pérdida de localización (box)	0.05
Ganancia de pérdida de clasificación (cls)	0.5
Peso positivo de clasificación (cls_pw)	1.0
Ganancia de pérdida de objetividad (obj)	1.0
Peso positivo de objetividad (obj_pw)	1.0
Umbral IoU de entrenamiento (iou_t)	0.20
Umbral de relación de anclas (anchor_t)	4.0
Variación de tonalidad (hsv_h)	0.015
Variación de saturación (hsv_s)	0.7
Variación de brillo (hsv_v)	0.4
Perspectiva (perspective)	0.0
Volteo vertical (flipud)	0.0
Volteo horizontal (fliplr)	0.5
Mosaic (mosaic)	1.0

Los parámetros `box`, `cls` y `obj` controlaron la contribución de las pérdidas de localización, clasificación y objetividad, respectivamente. Los valores `iou_t` y `anchor_t` intervinieron en la asignación de los objetos reales a las predicciones y anclas utilizadas durante el entrenamiento. Las variaciones HSV, el volteo horizontal y Mosaic se mantuvieron activos en los cuatro experimentos; por ello, la denominación *baseline* no significa ausencia total de aumento de datos, sino el uso de la configuración base frente a la cual se compararon las demás estrategias.

3.5.5. Configuraciones experimentales

Experimento 1: YOLOv5L con configuración base

El primer experimento estableció la línea base para la comparación. Se utilizó el optimizador de descenso de gradiente estocástico con momento (SGD), una tasa de aprendizaje inicial de 0.01, un momento de 0.937 y un decaimiento de pesos de 0.0005. Las primeras tres épocas funcionaron como periodo de calentamiento gradual (*warm-up*), con la finalidad de reducir actualizaciones inestables al comienzo del entrenamiento.

Además de las transformaciones comunes, se aplicaron traslaciones de 0.1 y un escalado de 0.5. No se utilizaron rotaciones, cizallamiento, Focal Loss, MixUp ni Copy-Paste. Esta configuración proporcionó el punto de referencia para determinar el efecto de las modificaciones incorporadas en los experimentos siguientes.

Tabla 3.6

Configuración específica del Experimento 1

Parámetro	Valor	Parámetro	Valor
Optimizador	SGD	lr0	0.01
lrf	0.1	momentum	0.937
weight_decay	0.0005	warmup_epochs	3.0
fl_gamma	0.0	degrees	0.0
translate	0.1	scale	0.5
shear	0.0	mixup	0.0
copy_paste	0.0		

Experimento 2: YOLOv5L con Focal Loss

El segundo experimento mantuvo la configuración del Experimento 1 y modificó el parámetro `fl_gamma` de 0.0 a 2.0. Focal Loss reduce la contribución de los ejemplos que el modelo resuelve con facilidad y concentra una mayor proporción del aprendizaje en los ejemplos difíciles. El valor $\gamma = 2,0$ determinó la intensidad con la que se redujo el peso de los ejemplos fáciles.

El objetivo fue comprobar si esta modificación mejoraba la detección de muestras

difíciles o con menor representación. Debido a que los demás parámetros se mantuvieron con respecto a la línea base, las variaciones de desempeño entre los experimentos 1 y 2 pudieron asociarse principalmente a la incorporación de Focal Loss.

Tabla 3.7

Configuración específica del Experimento 2

Parámetro	Valor	Parámetro	Valor
Optimizador	SGD	lr0	0.01
lrf	0.1	momentum	0.937
weight_decay	0.0005	warmup_epochs	3.0
fl_gamma	2.0	degrees	0.0
translate	0.1	scale	0.5
shear	0.0	mixup	0.0
copy_paste	0.0		

Experimento 3: YOLOv5L con aumento de datos avanzado

El tercer experimento evaluó una configuración de aumento de datos avanzado. Se conservaron el optimizador SGD, la tasa de aprendizaje inicial de 0.01, el momento de 0.937, el decaimiento de pesos de 0.0005 y el calentamiento gradual durante las primeras tres épocas. Asimismo, se mantuvo `fl_gamma=0.0`; por tanto, no se utilizó Focal Loss.

Con respecto a la línea base, se incorporaron rotaciones de hasta 10 grados y un cizallamiento de 2.0; además, la traslación se incrementó de 0.1 a 0.2 y el escalado de 0.5 a 0.9. También se configuraron MixUp con un valor de 0.15 y Copy-Paste con un valor de 0.1. Estas transformaciones se generaron dinámicamente durante el entrenamiento y no modificaron permanentemente las imágenes originales.

La finalidad de esta estrategia fue exponer al modelo a una mayor diversidad de orientaciones, escalas y disposiciones espaciales de los defectos, disminuyendo su dependencia de condiciones visuales específicas del conjunto de entrenamiento.

Tabla 3.8*Configuración específica del Experimento 3*

Parámetro	Valor	Parámetro	Valor
Optimizador	SGD	lr0	0.01
lrf	0.1	momentum	0.937
weight_decay	0.0005	warmup_epochs	3.0
fl_gamma	0.0	degrees	10.0
translate	0.2	scale	0.9
shear	2.0	mixup	0.15
copy_paste	0.1		

Experimento 4: YOLOv5L con AdamW y Cosine Annealing

El cuarto experimento evaluó el optimizador AdamW junto con una programación cosenoidal de la tasa de aprendizaje. AdamW adapta las actualizaciones de los parámetros y aplica el decaimiento de pesos de forma desacoplada. En esta configuración, la tasa de aprendizaje inicial se redujo a 0.001, el factor final `lrf` se estableció en 0.01, el decaimiento de pesos se incrementó a 0.05 y el periodo de calentamiento se amplió a cinco épocas.

La programación *Cosine Annealing*, habilitada en el comando de entrenamiento mediante `--cos-lr`, redujo gradualmente la tasa de aprendizaje siguiendo una curva cosenoidal hasta aproximarse a 0,00001. No se utilizó Focal Loss y se conservaron las transformaciones de aumento correspondientes a la configuración base. El propósito fue analizar si esta combinación mejoraba la estabilidad del entrenamiento y la capacidad de generalización del modelo.

Tabla 3.9*Configuración específica del Experimento 4*

Parámetro	Valor	Parámetro	Valor
Optimizador	AdamW	lr0	0.001
lrf	0.01	momentum	0.937
weight_decay	0.05	warmup_epochs	5.0
fl_gamma	0.0	degrees	0.0
translate	0.1	scale	0.5
shear	0.0	mixup	0.0
copy_paste	0.0	Programación cosenoidal	Sí

3.5.6. Criterio de evaluación y selección

El desempeño de los modelos se evaluó mediante precisión, *recall*, mAP@0.5 y mAP@0.5:0.95. La precisión representa la proporción de detecciones positivas que correspondieron a defectos reales, mientras que el *recall* indica la proporción de defectos existentes que fueron detectados por el modelo.

La métrica mAP@0.5 resume la precisión promedio de las clases cuando una predicción se considera correctamente localizada a partir de un IoU de 0.5. Por su parte, mAP@0.5:0.95 promedia el desempeño en umbrales IoU desde 0.5 hasta 0.95, con incrementos de 0.05, por lo que constituye una evaluación más estricta de la calidad de las cajas delimitadoras.

La selección de la mejor configuración se realizó exclusivamente con los resultados del conjunto de validación. El conjunto de prueba no intervino en el ajuste de hiperparámetros ni en la decisión del modelo ganador; se utilizó después de concluir la selección para informar el desempeño final y efectuar una comparación descriptiva.

3.6. Resultados de la evaluación experimental

3.6.1. Comparación en el conjunto de validación

La Tabla 3.10 presenta los resultados promedio obtenidos por cada configuración en el conjunto de validación. La columna de épocas indica la cantidad ejecutada antes de que finalizara cada entrenamiento.

Tabla 3.10*Comparación de los experimentos en el conjunto de validación*

Exp.	Precisión	Recall	mAP@0.5	mAP@0.5:0.95	Épocas
1	0.861	0.798	0.832	0.467	42
2	0.914	0.714	0.789	0.467	59
3	0.938	0.826	0.898	0.488	78
4	0.824	0.716	0.775	0.418	35

El Experimento 1 alcanzó una precisión de 0.861, un *recall* de 0.798, un mAP@0.5 de 0.832 y un mAP@0.5:0.95 de 0.467. Estos valores establecieron el punto de referencia para las demás configuraciones. El Experimento 2 incrementó la precisión hasta 0.914, pero redujo el *recall* a 0.714 y el mAP@0.5 a 0.789. En consecuencia, Focal Loss produjo detecciones más conservadoras, sin generar una mejora general frente a la línea base.

El Experimento 3 obtuvo los valores más altos en las cuatro métricas: precisión de 0.938, *recall* de 0.826, mAP@0.5 de 0.898 y mAP@0.5:0.95 de 0.488. En comparación con el Experimento 1, presentó incrementos de 0.077 en precisión, 0.028 en *recall*, 0.066 en mAP@0.5 y 0.021 en mAP@0.5:0.95. Estos resultados indican que la configuración de aumento avanzado proporcionó el mejor equilibrio entre detección y localización dentro del conjunto de validación.

El Experimento 4 registró los valores más bajos de mAP@0.5 y mAP@0.5:0.95, con 0.775 y 0.418, respectivamente. En las condiciones evaluadas, AdamW y la programación cosenoidal no superaron a SGD. Este resultado únicamente permite concluir que la configuración concreta utilizada no produjo una mejora; no descarta que otros valores de AdamW puedan alcanzar un desempeño diferente.

3.6.2. Selección del modelo con mejor desempeño

El Experimento 3 fue seleccionado como la configuración ganadora debido a que alcanzó simultáneamente la mayor precisión, el mayor *recall*, el mejor mAP@0.5 y el mejor

mAP@0.5:0.95 en validación. La selección no se realizó porque un único indicador resultara favorable, sino por la superioridad conjunta del modelo en las cuatro métricas consideradas.

En consecuencia, los pesos `best.pt` obtenidos en el Experimento 3 fueron adoptados como modelo final para su evaluación confirmatoria y su posterior integración en el prototipo de detección en tiempo real.

3.6.3. Comparación en el conjunto de prueba

Una vez concluida la selección mediante validación, los archivos `best.pt` de las cuatro configuraciones fueron evaluados en el conjunto de prueba con fines descriptivos. Estos resultados no se utilizaron para modificar los hiperparámetros, repetir entrenamientos ni cambiar el modelo seleccionado. El desempeño final del modelo de la investigación corresponde al Experimento 3.

Resultados del Experimento 1

Tabla 3.11

Resultados del Experimento 1 en el conjunto de prueba

Clase	Precisión	Recall	mAP@0.5	mAP@0.5:0.95
<i>gap</i>	0.971	0.969	0.976	0.534
<i>running-stitch</i>	0.992	0.957	0.975	0.668
Promedio	0.981	0.963	0.976	0.601

Resultados del Experimento 2

Tabla 3.12

Resultados del Experimento 2 en el conjunto de prueba

Clase	Precisión	Recall	mAP@0.5	mAP@0.5:0.95
<i>gap</i>	0.962	0.941	0.975	0.526
<i>running-stitch</i>	0.976	0.928	0.960	0.693
Promedio	0.969	0.934	0.967	0.609

Resultados del Experimento 3

Tabla 3.13

Resultados del Experimento 3 en el conjunto de prueba

Clase	Precisión	Recall	mAP@0.5	mAP@0.5:0.95
<i>gap</i>	0.967	0.978	0.991	0.646
<i>running-stitch</i>	1.000	0.954	0.989	0.704
Promedio	0.983	0.966	0.990	0.675

Resultados del Experimento 4

Tabla 3.14

Resultados del Experimento 4 en el conjunto de prueba

Clase	Precisión	Recall	mAP@0.5	mAP@0.5:0.95
<i>gap</i>	0.926	0.971	0.979	0.527
<i>running-stitch</i>	0.999	0.928	0.959	0.614
Promedio	0.963	0.949	0.969	0.570

La Tabla 3.15 sintetiza los promedios de las cuatro configuraciones en el conjunto de prueba.

Tabla 3.15

Comparación promedio de los experimentos en el conjunto de prueba

Exp.	Precisión	Recall	mAP@0.5	mAP@0.5:0.95
1	0.981	0.963	0.976	0.601
2	0.969	0.934	0.967	0.609
3	0.983	0.966	0.990	0.675
4	0.963	0.949	0.969	0.570

El Experimento 3 también obtuvo los mayores promedios en prueba: precisión de 0.983, *recall* de 0.966, mAP@0.5 de 0.990 y mAP@0.5:0.95 de 0.675. En comparación con la línea base, el mAP@0.5 aumentó 0.014 y el mAP@0.5:0.95 se incrementó 0.074. La mejora en la métrica más estricta evidencia una delimitación más precisa de las imperfecciones.

En el análisis por clase, el Experimento 3 alcanzó un mAP@0.5:0.95 de 0.646 para *gap*, frente a 0.534 en la línea base. Para *running-stitch*, esta métrica aumentó de 0.668 a 0.704.

La precisión de 1.000 obtenida para *running-stitch* indica que, bajo el umbral empleado en la evaluación, la proporción de falsos positivos fue prácticamente nula; sin embargo, el *recall* de 0.954 muestra que todavía existieron algunos defectos no detectados.

Los valores del conjunto de prueba fueron superiores a los de validación. Esta diferencia no implica que el modelo haya mejorado después del entrenamiento, pues se utilizaron los mismos pesos. Puede estar relacionada con diferencias en la composición o dificultad de ambos subconjuntos y, por ello, debe interpretarse considerando la variabilidad del conjunto de datos.

3.6.4. Comparación global de los experimentos

La Tabla 3.16 integra el mAP@0.5 de validación y prueba, el número de épocas ejecutadas y el tiempo de entrenamiento de cada configuración.

Tabla 3.16

Comparación global de los experimentos

Exp.	Estrategia	mAP@0.5 Val.	mAP@0.5 Prueba	Épocas	Tiempo
1	Configuración base	0.832	0.976	42	0.547 h
2	Focal Loss	0.789	0.967	59	0.800 h
3	Aumento avanzado	0.898	0.990	78	1.059 h
4	AdamW + Cosine Annealing	0.775	0.969	35	0.452 h

El Experimento 3 requirió el mayor número de épocas y el mayor tiempo de entrenamiento; no obstante, alcanzó el mAP@0.5 más alto en validación y en prueba. Este resultado muestra que, dentro de las configuraciones evaluadas, el aumento de datos avanzado ofreció el mejor equilibrio entre capacidad de detección y costo computacional. El Experimento 4 finalizó en menos tiempo, pero obtuvo un desempeño inferior, por lo que la reducción del costo de entrenamiento no compensó la pérdida de precisión observada.

3.6.5. Análisis del modelo seleccionado

Matriz de confusión

La matriz de confusión del Experimento 3 permite analizar los aciertos y errores de clasificación de las clases *gap* y *running-stitch*, así como su relación con el fondo. Esta representación complementa las métricas globales al mostrar la distribución de las predicciones correctas, los falsos positivos y los defectos no detectados.

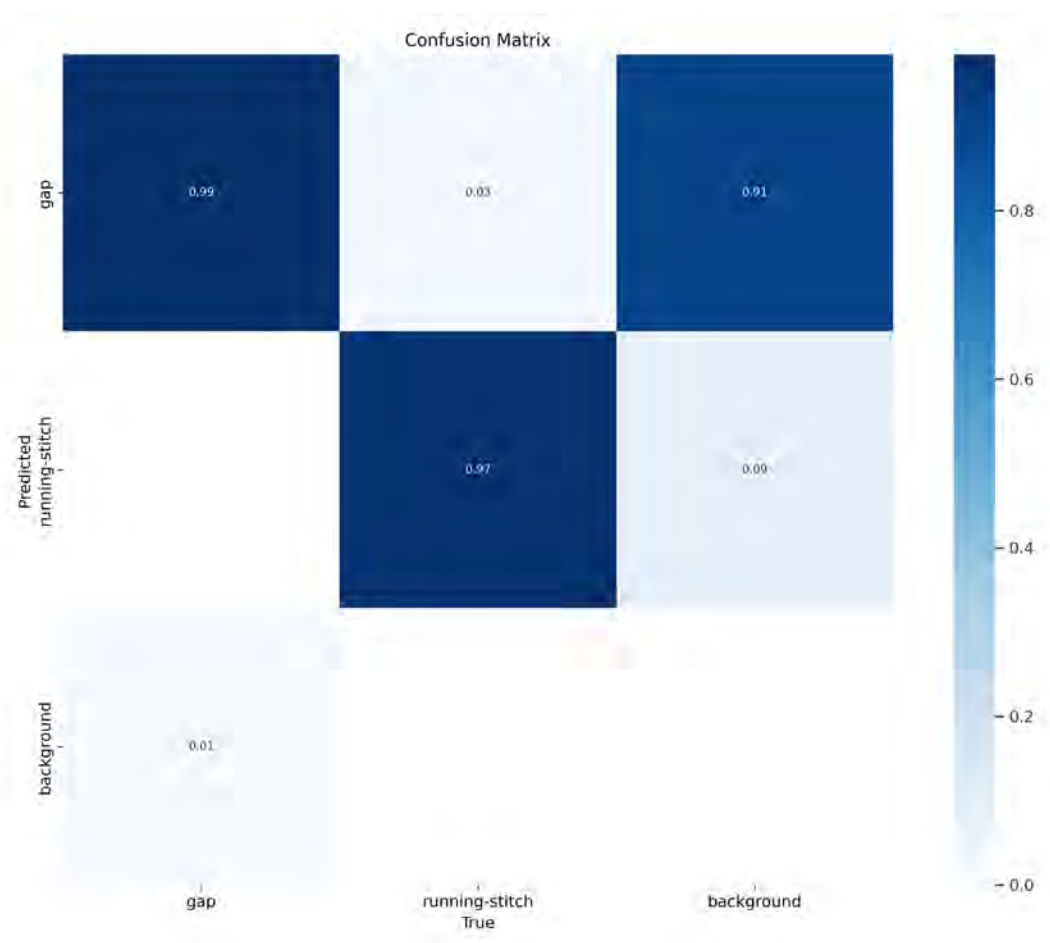


Figura 3.1

Matriz de confusión del Experimento 3 en el conjunto de prueba

Curva Precision–Recall

La curva Precision–Recall muestra el comportamiento del modelo ante distintos umbrales de confianza y representa la relación entre la proporción de detecciones correctas y la capacidad para recuperar los defectos existentes. En la Figura 3.2 se observa que la curva

se mantiene próxima al valor máximo de precisión durante un intervalo amplio de *recall*. La disminución más pronunciada aparece cuando el *recall* se aproxima a 1.0, debido a que la reducción del umbral de confianza permite predicciones más permisivas.

El mAP@0.5 de 0.990 resume el desempeño de las dos clases con un criterio de coincidencia IoU de 0.5. Por otro lado, el mAP@0.5:0.95 de 0.675 evalúa la clasificación y la localización mediante criterios progresivamente más estrictos.

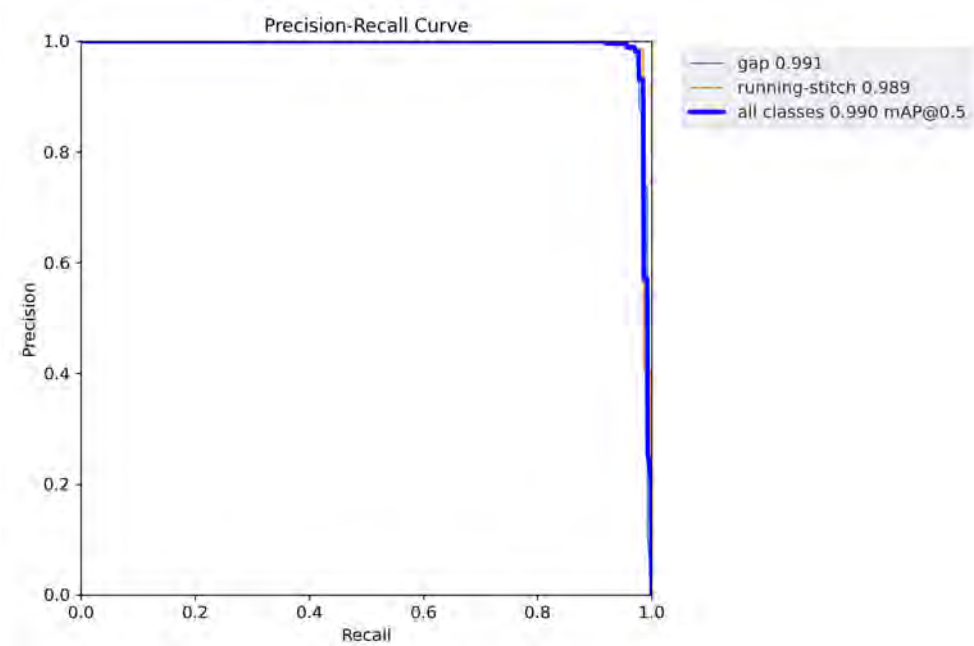


Figura 3.2

Curva Precision–Recall del Experimento 3 en el conjunto de prueba

Curva F1–Confidence

La curva F1–Confidence representa la variación del valor F1 en función del umbral de confianza. El F1-score corresponde a la media armónica entre precisión y *recall*, por lo que permite evaluar el equilibrio entre ambas métricas.

En la Figura 3.3 se observa que el valor F1 aumenta durante los primeros niveles de confianza y se mantiene elevado en un intervalo amplio. El valor máximo, cercano a 0.97, se alcanza aproximadamente con un umbral de confianza de 0.71. A partir de umbrales superiores, el F1 disminuye debido a la reducción del *recall*, ya que el modelo adopta un

criterio más estricto para aceptar sus predicciones. En consecuencia, el valor aproximado de 0.71 constituye un punto de referencia para configurar el umbral de operación del prototipo, aunque su elección definitiva debe considerar el costo relativo de omitir un defecto frente al de generar una falsa alarma.

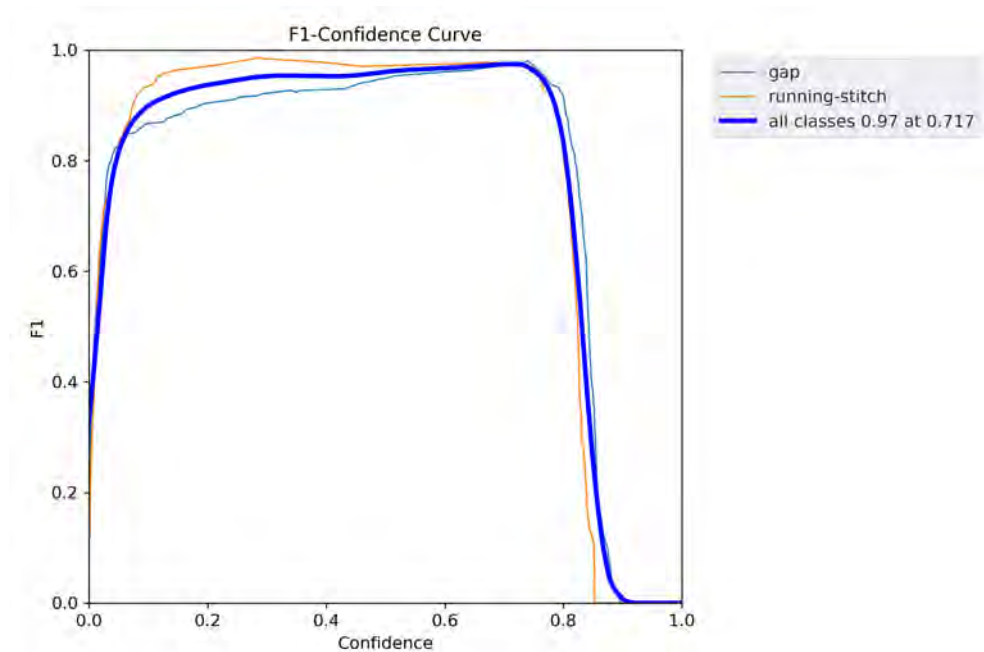


Figura 3.3

Curva F1-Confidence del Experimento 3 en el conjunto de prueba

3.6.6. Resultados visuales del modelo seleccionado

Como complemento del análisis cuantitativo, se realizó una inspección cualitativa de imágenes del conjunto de validación. Para ello, se compararon ejemplos de las anotaciones de referencia contenidas en `val_batch1_labels.jpg` con las predicciones generadas por el modelo en `val_batch1_pred.jpg`.

La Figura 3.4 presenta las cajas delimitadoras anotadas manualmente en una muestra del conjunto de validación.

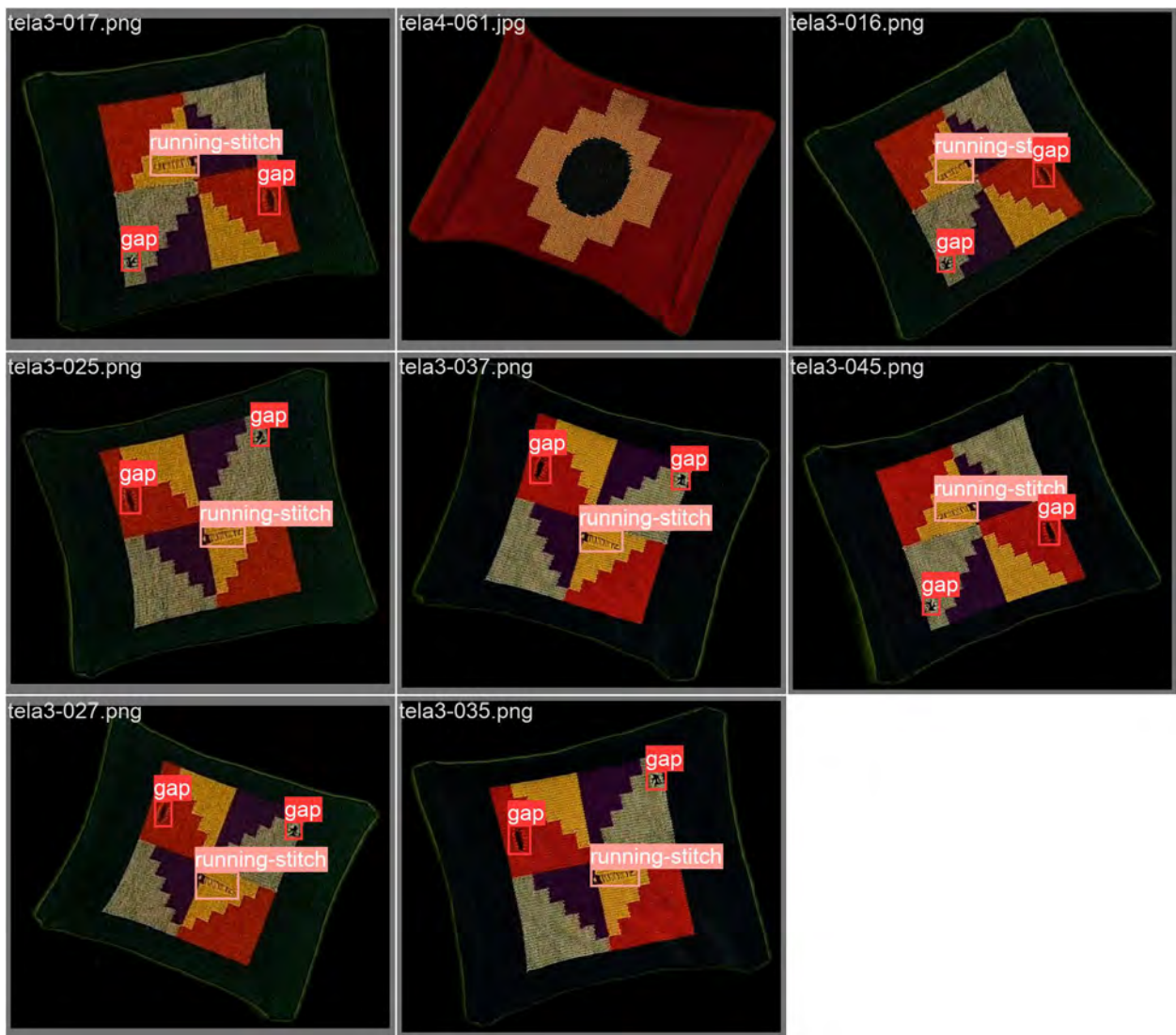


Figura 3.4

Anotaciones de referencia de una muestra del conjunto de validación

La Figura 3.5 muestra las predicciones realizadas por el Experimento 3 sobre las mismas imágenes. Se observan coincidencias con las regiones anotadas y algunas detecciones adicionales en otras zonas del tejido. Estas detecciones deben revisarse individualmente para determinar si corresponden a falsos positivos o a imperfecciones que no fueron incluidas durante la anotación manual.

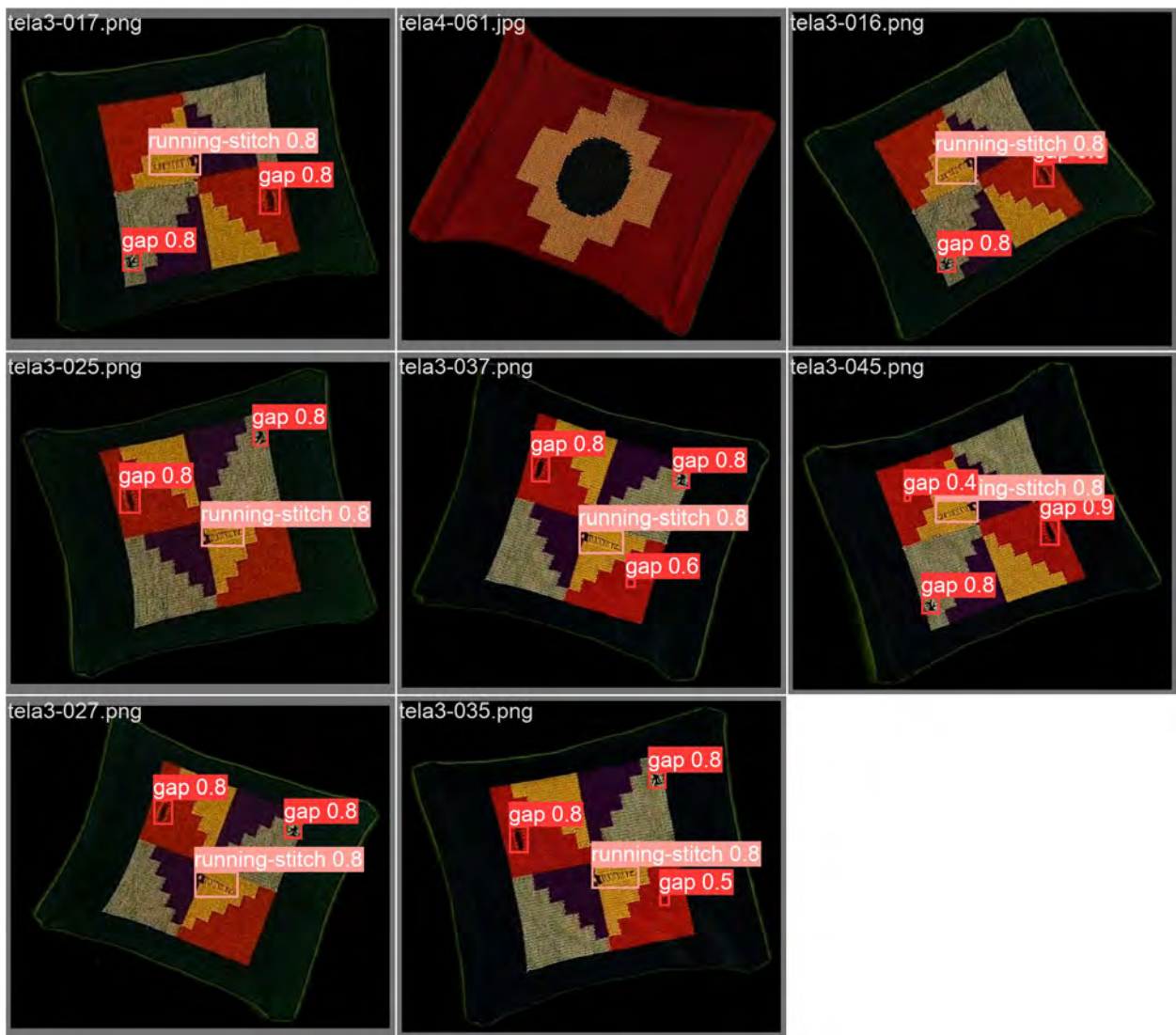


Figura 3.5

Predicciones del Experimento 3 sobre una muestra del conjunto de validación

3.6.7. Almacenamiento e integración del modelo final

El archivo `best.pt` del Experimento 3 fue seleccionado como modelo final de la investigación. Este archivo contiene los pesos correspondientes a la época con mejor aptitud en validación y fue utilizado en la integración del módulo de detección en tiempo real. La elección de estos pesos permitió trasladar al prototipo la configuración que obtuvo el mejor desempeño durante la comparación experimental.

3.7. Implementación del prototipo de detección en tiempo real

El modelo seleccionado se integró en un prototipo de detección en tiempo real desarrollado en Python con PyTorch y OpenCV. El flujo de ejecución comprende la carga de los pesos entrenados, la apertura de una fuente de video, la lectura de cuadros, la inferencia, el filtrado por confianza, la visualización de los resultados y la liberación de los recursos de captura.

3.7.1. Flujo de desarrollo del prototipo

El desarrollo del prototipo se realizó a partir de los pesos `best.pt` obtenidos en el Experimento 3, correspondiente al modelo YOLOv5L con aumento de datos avanzado. En primer lugar, se definieron los requisitos funcionales del sistema y se configuró el entorno de ejecución mediante Python, PyTorch y OpenCV. Posteriormente, el modelo seleccionado se integró con la captura de video mediante cámara, permitiendo procesar los fotogramas, identificar huecos y puntos corridos y visualizar los resultados mediante cuadros delimitadores, etiquetas de clase, niveles de confianza y velocidad de procesamiento. Finalmente, se realizaron pruebas funcionales y ajustes bajo condiciones controladas hasta obtener el prototipo implementado. La Figura 3.6 presenta el flujo seguido.

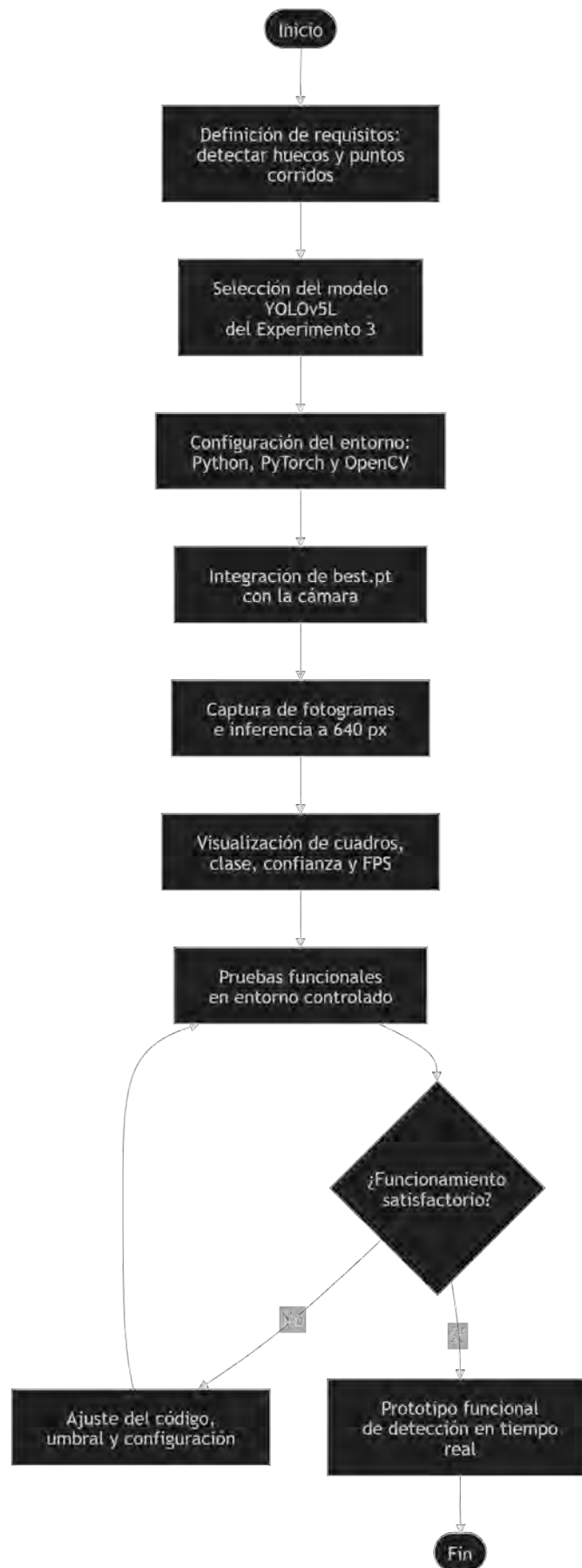


Figura 3.6

Diagrama de flujo del desarrollo del prototipo de detección en tiempo real.

El prototipo acepta mediante argumentos la ruta del modelo, la fuente de video, el umbral de confianza, la resolución solicitada, el tamaño de inferencia, el número de cuadros omitidos y el backend de captura. Esta parametrización permite adaptar la ejecución sin modificar el código. El umbral de confianza se fijó por defecto en 0.50, por lo que las predicciones inferiores a dicho valor no se visualizan.

El modelo personalizado se carga mediante `torch.hub.load` y se ejecuta en CUDA cuando existe una GPU compatible; en caso contrario, utiliza la CPU. Para cada inferencia se obtienen las coordenadas (x_1, y_1, x_2, y_2) , la confianza y el identificador de clase. OpenCV dibuja el cuadro delimitador y una etiqueta con el nombre de la clase y su confianza, se puede ver una demostración en las figuras 3.7. La supresión no máxima forma parte del proceso de salida de YOLOv5 y evita conservar múltiples cuadros redundantes para una misma región.

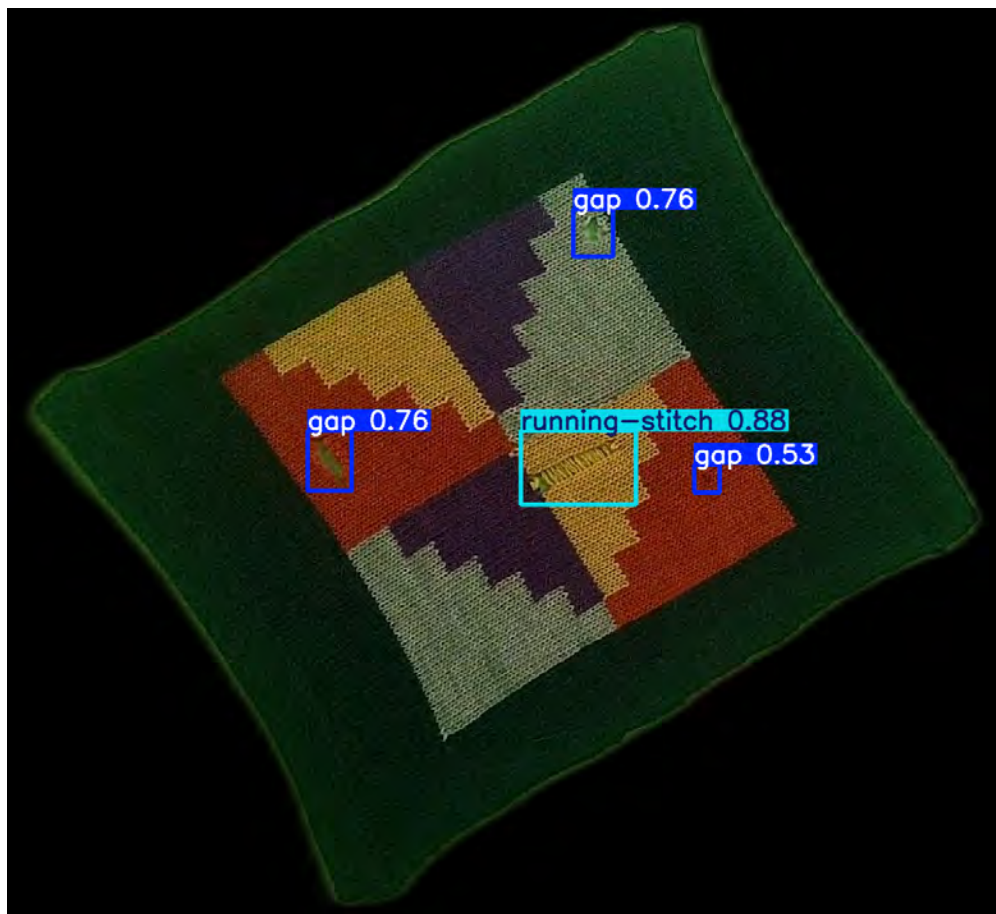
El parámetro `infer-size` el modelo fue evaluado a 1024 píxeles. Reducir el tamaño de inferencia o omitir cuadros puede aumentar los FPS, pero también puede disminuir la capacidad de localizar imperfecciones pequeñas.

La velocidad se estimó mediante el inverso del tiempo transcurrido entre cuadros y se estabilizó con un promedio móvil de las últimas quince mediciones. Este valor describe la velocidad de visualización del prototipo; para afirmar un rendimiento determinado deben reportarse el FPS promedio observado, la resolución real obtenida y el equipo de inferencia. La prueba realizada demuestra la integración funcional bajo condiciones controladas, pero no mide todavía su impacto sobre tiempos de inspección, retrabajos o costos de una línea industrial.

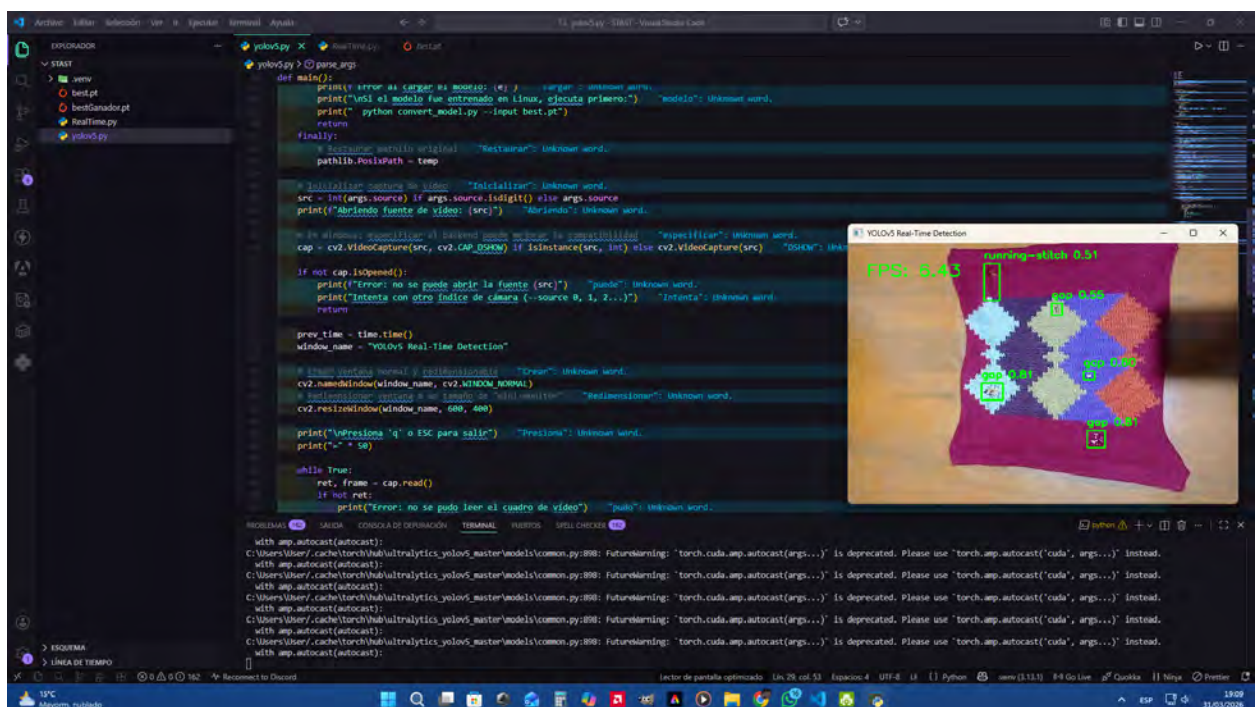
El código completo se presenta en el Anexo B.

Figura 3.7

Captura en tiempo real de imperfecciones en mantas de alpaca



(a) *Detección en tiempo real - escenario 1*



(b) *Detección en tiempo real - escenario 2*

Entorno de entrenamiento e inferencia

El entrenamiento de los cuatro experimentos se realizó en Google Colab mediante una unidad gráfica [*modelo exacto de GPU*]. Se utilizaron Python, PyTorch y el repositorio oficial de YOLOv5. Para favorecer la reproducibilidad, se registró el identificador [*commit exacto*] del repositorio, así como las versiones de Python, PyTorch y CUDA.

El equipo personal con procesador AMD Ryzen 7 5700G y 16 GB de memoria RAM se utilizó para la preparación de archivos, revisión de resultados y desarrollo del prototipo, pero no para el entrenamiento principal de los modelos.

La prueba funcional de inferencia mediante cámara en vivo se realizó en una computadora portátil con [*procesador, GPU y RAM*]. La cámara operó a una resolución de [*resolución*], mientras que el tamaño de entrada para la inferencia fue de 640 píxeles. Bajo esta configuración se observó una velocidad promedio aproximada de 40 FPS. Este valor corresponde a la ejecución del prototipo bajo condiciones controladas y no constituye una medición de rendimiento dentro de una línea industrial.

3.8. Limitaciones del estudio

A pesar de los resultados obtenidos, la presente investigación presenta ciertas limitaciones que deben ser consideradas al momento de interpretar los resultados.

En primer lugar, el conjunto de datos fue recolectado bajo condiciones controladas de iluminación y con un fondo uniforme, lo cual podría no representar completamente las condiciones reales de un entorno industrial, donde existen variaciones de luz, ruido visual y fondos no homogéneos.

Asimismo, el número de muestras (549 imágenes) puede considerarse reducido para aplicaciones a gran escala, lo que podría afectar la capacidad del modelo ante nuevos tipos de tejidos o imperfecciones no contemplados en el entrenamiento.

Finalmente, el entrenamiento del modelo fue realizado en un entorno computacional con recursos limitados, lo cual restringe la experimentación con arquitecturas más complejas o conjuntos de datos de mayor tamaño.

Estas limitaciones abren la posibilidad de futuras investigaciones orientadas a mejorar la robustez y aplicabilidad del prototipo en entornos industriales reales.

Capítulo 4

Análisis y discusión de resultados

4.1. Análisis de resultados respecto a los objetivos

El objetivo general de la presente investigación fue implementar un módulo de visión computacional para la detección de imperfecciones en mantas de tejido punto elaboradas con fibras de alpaca. A partir de los resultados obtenidos en los experimentos realizados, se puede afirmar que dicho objetivo fue alcanzado satisfactoriamente, dado que se logró desarrollar, entrenar y validar un modelo de detección automática de imperfecciones textiles basado en la arquitectura YOLOv5L, el cual fue posteriormente integrado en un prototipo de detección en tiempo real.

En relación con el primer objetivo específico, referido a la construcción de un conjunto de datos de imágenes de mantas de tejido punto de alpaca, se logró conformar un dataset propio debidamente etiquetado, el cual permitió entrenar y validar modelos de visión computacional orientados a la detección de las imperfecciones más comunes: huecos y puntos corridos. La creación de este conjunto de datos resulta especialmente relevante debido a que, durante la revisión bibliográfica realizada, no se identificaron conjuntos de datos públicos específicamente orientados a imperfecciones en mantas de tejido punto elaboradas

con fibras de alpaca, lo que constituye un aporte significativo para futuras investigaciones en el ámbito del control de calidad textil artesanal e industrial.

Respecto al segundo objetivo específico, orientado al desarrollo de modelos de visión computacional para identificar imperfecciones de manufactura, los resultados experimentales muestran que el Experimento 3, correspondiente a YOLOv5L con aumento de datos avanzado, alcanzó el mejor desempeño entre las configuraciones evaluadas. En el conjunto de validación obtuvo una precisión de 0.938, un recall de 0.826, un mAP@50 de 0.898 y un mAP@50–95 de 0.488. Posteriormente, en el conjunto de prueba alcanzó una precisión de 0.983, un recall de 0.966, un mAP@50 de 0.990 y un mAP@50–95 de 0.675. La precisión obtenida indica una baja proporción de falsas detecciones, mientras que el recall refleja una elevada capacidad para identificar las imperfecciones presentes. Por su parte, los valores de mAP evalúan conjuntamente la clasificación y la localización de las imperfecciones bajo diferentes criterios de superposición. Estos resultados sugieren que la configuración con aumento de datos presentó una mejor capacidad de generalización dentro de las condiciones evaluadas.

En cuanto al tercer objetivo específico, referido a integrar el modelo de visión computacional seleccionado en un prototipo de detección mediante cámara en vivo para apoyar la identificación de huecos y puntos corridos durante la etapa de inspección visual final, se implementó un prototipo utilizando los pesos `best.pt` obtenidos en el Experimento 3. El prototipo procesó en tiempo real las imágenes captadas por la cámara de una computadora portátil y permitió visualizar los cuadros delimitadores (*bounding boxes*), la clase de la imperfección detectada (*gap* o *running-stitch*) y el nivel de confianza correspondiente a cada predicción. Durante las pruebas realizadas bajo condiciones controladas, el prototipo alcanzó una velocidad aproximada de 40 cuadros por segundo (FPS), lo que permitió mantener una visualización fluida de las detecciones. De este modo, se cumplió el tercer objetivo específico al integrar funcionalmente el modelo seleccionado en un prototipo capaz de apoyar la

identificación de huecos y puntos corridos mediante una cámara en vivo. No obstante, el prototipo constituye una herramienta de apoyo y no reemplaza por completo la decisión del responsable de la inspección. Asimismo, su desempeño y su contribución a la reducción de los tiempos de inspección, errores y retrabajos deberán evaluarse posteriormente mediante pruebas piloto durante la etapa de inspección visual final en un entorno productivo real.

4.2. Discusión de resultados respecto a los antecedentes

Los resultados del Experimento 3 muestran que YOLOv5L con aumento de datos avanzado fue capaz de detectar huecos y puntos corridos en las mantas incluidas en el conjunto de prueba. El modelo alcanzó una precisión de 0.983, un *recall* de 0.966, un mAP@50 de 0.990 y un mAP@50–95 de 0.675. La diferencia entre mAP@50 y mAP@50–95 indica que el desempeño fue muy elevado con un criterio de superposición de 0.50, mientras que la localización disminuyó al aplicar umbrales progresivamente más estrictos. Por consiguiente, aunque el modelo identificó la mayoría de las imperfecciones, todavía existe margen para mejorar el ajuste preciso de los cuadros delimitadores.

El estudio de Hoyos Montes [2020] también evidencia la utilidad del aprendizaje automático para la inspección textil. Sin embargo, su precisión reportada y la obtenida en esta investigación no constituyen una comparación directa, debido a que se emplearon conjuntos de datos, clases, arquitecturas y protocolos de evaluación diferentes. La relación entre ambos trabajos se encuentra en que desarrollan soluciones especializadas a partir de imágenes del contexto textil; el aporte particular del presente estudio consiste en abordar mantas de tejido punto elaboradas con fibras de alpaca mediante un conjunto de datos propio.

La red en cascada propuesta por Zhang [2021] se orienta a localizar regiones anómalas en telas con patrones complejos mediante acercamientos sucesivos. Aunque su arquitectura y su conjunto de datos difieren de los del presente trabajo, ambos estudios coinciden en la necesidad de conservar información espacial suficiente para reconocer regiones pequeñas dentro de una textura repetitiva. En esta investigación, dicha necesidad se abordó mediante YOLOv5L, una resolución de entrada de 1024×1024 píxeles y aumento de datos orientado a variaciones de escala y posición.

Por su parte, Wei et al. [2021] utilizaron mecanismos de atención multinivel para el reconocimiento de imperfecciones textiles pequeñas y coexistentes. El presente estudio no incorporó módulos de atención, por lo que no corresponde atribuirle las mismas propiedades ni comparar directamente sus métricas. No obstante, la detección de varias instancias de huecos y puntos corridos en una imagen confirma la pertinencia de utilizar características multiescala para este tipo de problema. En YOLOv5L, la fusión de características de diferentes niveles y el aumento de datos contribuyeron a representar variaciones de tamaño y ubicación dentro de las condiciones evaluadas.

El enfoque no supervisado de Koulali and Eskill [2023] busca reducir la dependencia de grandes volúmenes de anotaciones. Esta necesidad también fue evidente en la presente investigación, pues fue necesario construir y etiquetar un conjunto propio de 549 imágenes. A diferencia de dicho antecedente, aquí se empleó aprendizaje supervisado para distinguir explícitamente dos clases. La comparación metodológica permite identificar una línea futura de trabajo: incorporar aprendizaje semisupervisado o no supervisado para ampliar el conjunto de datos sin incrementar proporcionalmente el esfuerzo de etiquetado.

En conjunto, los antecedentes permiten interpretar los resultados, pero no establecer cuál propuesta es superior, puesto que ninguna fue entrenada y evaluada sobre el mismo conjunto de datos y bajo un protocolo común. La evidencia obtenida permite concluir úni-

camente que, dentro del dataset y de las condiciones controladas de esta investigación, YOLOv5L con aumento de datos avanzado fue la mejor de las cuatro configuraciones evaluadas. Su contribución reside en la aplicación específica a mantas de alpaca, la construcción del conjunto de datos y la integración del modelo en un prototipo de detección mediante cámara en vivo.

Conclusiones

A partir de los resultados obtenidos en la presente investigación, se concluye lo siguiente:

- En relación con el primer objetivo específico, se logró construir un conjunto de datos propio de imágenes de mantas de tejido punto elaboradas con fibras de alpaca, debidamente etiquetado en función de las imperfecciones más frecuentes (huecos y puntos corridos). Este conjunto de datos, compuesto por un total de 549 imágenes, permitió entrenar y validar modelos de visión computacional, constituyéndose además en un aporte relevante debido a que, durante la revisión bibliográfica realizada, no se identificaron conjuntos de datos públicos específicamente orientados a imperfecciones en mantas de tejido punto elaboradas con fibras de alpaca.
- Respecto al segundo objetivo específico, se desarrollaron y evaluaron diversos modelos de detección de imperfecciones basados en la arquitectura YOLOv5, identificándose que el modelo YOLOv5L con técnicas avanzadas de aumento de datos alcanzó el mejor desempeño. En particular, en el conjunto de prueba se obtuvo una precisión de 0.983, un recall de 0.966, un mAP@50 de 0.990 y un mAP@50–95 de 0.675. Estos resultados evidencian la alta capacidad del modelo para detectar correctamente huecos y puntos corridos bajo las condiciones visuales representadas en el conjunto de datos, superando a las demás configuraciones evaluadas.

- En cuanto al tercer objetivo específico, se logró integrar el modelo seleccionado en un prototipo de detección en tiempo real y comprobar su funcionamiento mediante una cámara en vivo bajo condiciones controladas. El prototipo permitió visualizar los cuadros delimitadores, las clases detectadas, los niveles de confianza y la velocidad de procesamiento. Estos resultados demuestran la integración funcional y la factibilidad técnica del prototipo, el cual presenta potencial para apoyar el control de calidad de mantas de alpaca. Sin embargo, su impacto sobre la inspección manual, los retrabajos, los tiempos y los costos deberá evaluarse posteriormente mediante una prueba piloto en un entorno industrial real.
- Finalmente, se concluye que el objetivo general fue cumplido bajo las condiciones controladas del estudio, al implementarse un módulo funcional basado en YOLOv5L que alcanzó un mAP@50 de 0.990 en el conjunto de prueba y que fue integrado en un prototipo de detección mediante cámara en vivo.

Recomendaciones

A partir de los resultados obtenidos y las limitaciones identificadas durante el desarrollo de la presente investigación, se proponen las siguientes recomendaciones para trabajos futuros:

- Ampliar el conjunto de datos de imágenes incorporando nuevas muestras provenientes de distintos talleres textiles, regiones geográficas y condiciones de iluminación, con el fin de mejorar la capacidad de generalización del modelo frente a variaciones reales del entorno productivo.
- Incluir nuevos tipos de imperfecciones textiles en futuras investigaciones, tales como variaciones de densidad del tejido, manchas, deshilachados o deformaciones estructurales, con el objetivo de construir un prototipo de control de calidad más completo e integral.
- Evaluar arquitecturas más recientes de detección de objetos, como versiones posteriores de YOLO o modelos con mecanismos de atención, con el propósito de analizar posibles mejoras en la precisión de detección, especialmente en imperfecciones de tamaño reducido o con bajo contraste visual.
- Explorar enfoques de aprendizaje no supervisado o semi-supervisado para la detección de imperfecciones textiles, con el fin de reducir el esfuerzo de etiquetado manual requerido para la construcción de grandes conjuntos de datos, lo cual resulta particularmente

relevante en contextos donde no existen bases de datos especializadas.

- Implementar el prototipo propuesto en un entorno industrial real, integrándolo con cámaras de inspección en línea y prototipos de información de producción, con el objetivo de evaluar su impacto en términos de reducción de tiempos de inspección, disminución de retrabajos y mejora de la calidad del producto final.
- Desarrollar una interfaz de usuario amigable para operarios de planta, que permita visualizar las imperfecciones detectadas, registrar estadísticas de calidad por lote de producción y generar reportes automáticos que apoyen la toma de decisiones en el proceso de control de calidad.

Bibliografía

Alpaca del Perú. Características de la fibra de alpaca, 2025. URL <https://www.alpacadelperu.org.pe/caracteristicas-de-la-fibra-de-alpaca/>. Consultado el 8 de abril de 2025.

Diego Alonso Azabache-Alvarado, Jorge Armando Campero-Flórez, Alvaro Rodrigo Gallardo-Siguas, and Anthony Ramirez-Abad. Análisis de la evolución de la fibra de alpaca peruana del 2010 al 2019. Trabajo de investigación para optar el grado de bachiller en ingeniería industrial, Universidad de Lima, Lima, Perú, marzo 2021. URL <https://hdl.handle.net/20.500.12724/13275>.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, NY, 2006.

Sandy Black. *Knitting: Fashion, Industry, Craft*. V&A Publications, London, 2005.

Al C. Bovik, editor. *Handbook of Image and Video Processing*. Academic Press, Burlington, MA, 2005.

Jorge Calvo. Aumento de datos sintéticos. *European Valley*, may 2024. URL <https://europeanvalley.es/noticias/el-aumento-sintetico-de-datos/>. Profesor e Ingeniero Informático, Especialista en Inteligencia Artificial y Seguridad de la Información.

François Chollet. *Deep Learning with Python*. Manning Publications, Greenwich, CT, 2017.

- L. Cushman. Properties of alpaca fiber and its use in textile production. *Textile Research Journal*, 88(5):543–556, 2018.
- Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.
- S. García Jarpa and A. Halog. Pursuing a circular and sustainable textile industry in latin america. In *Sustainable Development in Latin America*, pages 101–115. Springer, 2021. doi: 10.1007/978-981-16-1850-5_6. URL https://link.springer.com/chapter/10.1007/978-981-16-1850-5_6.
- Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Pearson Education, Upper Saddle River, NJ, 3rd edition, 2009.
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, 2016.
- M. Gálvez and M. Hinojosa. Processing of alpaca fiber for textile use. *Textile Science & Engineering*, 47:95–102, 2011.
- Simon O. Haykin. *Neural Networks and Learning Machines*. Pearson, Upper Saddle River, NJ, 3rd edition, 2009.
- Roberto Hernández-Sampieri, César Fernández-Collado, and María del Pilar Baptista-Lucio. *Metodología de la investigación*. McGraw-Hill, México D.F., 6^a edition, 2014.
- June Hemmons Hiatt. *Los Principios de Tejer: Métodos y Técnicas de Tejer*. Simon and Schuster, New York, 1989. ISBN 978-0-671-55233-6.
- Yaqueline Aide Hoyos Montes. Detección de defectos en fibras textiles utilizando algoritmos de deep learning. Technical Report Informe 2020-01, Universidad de Antio-

quia, 2020. URL https://bibliotecadigital.udea.edu.co/bitstream/10495/15470/1/HoyosYaqueline_2020_DeteccionDefectosFibras.pdf

Junfeng Jing, Hongbo Yang, and Pengfei Li. Improved gabor filters for textile defect detection. *Procedia Engineering*, 15:1505–1512, 2011.

Asifullah Khan, Anabia Sohail, Umme Zahoora, and Aqsa Saeed Qureshi. A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53(8):5455–5516, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, volume 2, pages 1137–1143, Montreal, Canada, 1995.

Imane Koulali and M. Taner Eskil. Unsupervised textile defect detection using convolutional neural networks. *arXiv preprint arXiv:2312.00224*, 2023. URL <https://arxiv.org/pdf/2312.00224>.

M. Kumuthini, J. Shre Subasine, U. Veenaloshini, and D. Rajeswari. Fabric property and defect detection using deep learning model. *International Journal for Research in Applied Science and Engineering Technology*, 9:2345–2350, 2023. URL <https://www.ijraset.com/research-paper/fabric-property-and-defect-detection>.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

Yundong Li, Weigang Zhao, and Jiahao Pan. Deformable patterned fabric defect detection with fisher criterion-based deep learning. *IEEE Transactions on Industrial Informatics*,

- 14(2):1256–1264, 2017. doi: 10.1109/TASE.2016.2520955. URL <https://ieeexplore.ieee.org/document/7398188>.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.
- O. L. Lopera Lopera and J. Velez-Ocampo. Automation adoption in the textile industry of an emerging economy. In *The Palgrave Handbook of Corporate Sustainability in the Digital Era*, pages 55–73. Springer, 2021. doi: 10.1007/978-3-031-70789-6_9. URL https://link.springer.com/chapter/10.1007/978-3-031-70789-6_9.
- Miranda Miguel. La imagen digital. https://ve.scielo.org/scielo.php?script=sci_arttext&pid=S0016-35032009000200016, 2009.
- OpenCV. Smoothing images. https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html, 2024.
- Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2009.
- Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, Las Vegas, NV, 2016.
- K. Senthilkumar, K. Reena, D. Sathyapriya, and N. Vidhyasagar. Complex pattern jacquard fabrics defect detection using convolutional neural networks and multispectral imaging. *2024 2nd International Conference on Artificial Intelligence and Machine Learning Applications*, 2024. doi: 10.1109/AIMLA59606.2024.10531577. URL <https://ieeexplore.ieee.org/document/10531577>.

- Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.
- Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer Science & Business Media, London, 2010.
- Ultralytics. YOLOv5. <https://github.com/ultralytics/yolov5>, 2020. Accedido: 12 de septiembre de 2025.
- Ultralytics. YOLOv5 architecture description. <https://docs.ultralytics.com/yolov5/tutorials/architecture-description/>, 2026.
- Bing Wei, Kuangrong Hao, and Lei Gao. Mlma-net: Multi-level multi-attentional learning for multi-label object detection in textile defect images. *arXiv preprint arXiv:2102.00376*, 2021. URL <https://arxiv.org/pdf/2102.00376>.
- Zhiwei Zhang. Cascaded zoom-in network for patterned fabric defect detection. *arXiv preprint arXiv:2108.06760*, 2021. URL <https://arxiv.org/pdf/2108.06760>.

Apéndice A

Código de entrenamiento del Experimento 3 en Google Colab

En el presente anexo se incluye el código empleado para configurar, entrenar y evaluar el Experimento 3, correspondiente al modelo YOLOv5L con aumento de datos avanzado.

```
1 # Código para ejecutarse por celdas en Google Colab.
2 # Las líneas que comienzan con ! y % son comandos mágicos de Colab.
3
4 from google.colab import drive
5
6 drive.mount('/content/drive')
7
8 !cp '/content/drive/My Drive/Yolo/data.zip' '/content/data.zip'
9 !unzip -o '/content/data.zip' -d '/content/'
10
11 !git clone https://github.com/ultralytics/yolov5.git /content/yolov5
12 %cd /content/yolov5
13 %pip install -qr requirements.txt
14
15 # Registrar la versión exacta para asegurar la reproducibilidad.
16 !git rev-parse HEAD
17 !nvidia-smi
18
19 # =====
```

```

20 # EXPERIMENTO 3: YOLOv5L + AUMENTO DE DATOS AVANZADO
21 # =====
22
23 hyp_augment = """
24 lr0: 0.01
25 lrf: 0.1
26 momentum: 0.937
27 weight_decay: 0.0005
28 warmup_epochs: 3.0
29 warmup_momentum: 0.8
30 warmup_bias_lr: 0.1
31 box: 0.05
32 cls: 0.5
33 cls_pw: 1.0
34 obj: 1.0
35 obj_pw: 1.0
36 iou_t: 0.20
37 anchor_t: 4.0
38 fl_gamma: 0.0
39 hsv_h: 0.015
40 hsv_s: 0.7
41 hsv_v: 0.4
42 degrees: 10.0
43 translate: 0.2
44 scale: 0.9
45 shear: 2.0
46 perspective: 0.0
47 flipud: 0.0
48 fliplr: 0.5
49 mosaic: 1.0
50 mixup: 0.15
51 copy_paste: 0.1
52 """
53
54 # Se crea un único archivo en la raíz del repositorio clonado.
55 hyp_path = "/content/yolov5/hyp.augment.yaml"
56 with open(hyp_path, "w", encoding="utf-8") as f:
57     f.write(hyp_augment)

```

```

58
59 print(f"Archivo de hiperparámetros creado en: {hyp_path}")
60
61 print("\n" + "=" * 80)
62 print("EXPERIMENTO 3: YOLOv5L + aumento de datos avanzado")
63 print("=" * 80)
64
65 !WANDB_MODE=disabled python train.py \
66     --img 1024 \
67     --batch 8 \
68     --epochs 100 \
69     --data "/content/drive/My Drive/Yolo/config.yaml" \
70     --weights yolov5l.pt \
71     --optimizer SGD \
72     --hyp /content/yolov5/hyp.augment.yaml \
73     --patience 20 \
74     --name exp3_yolov5l_augment \
75     --cache \
76     --exist-ok
77
78 # Evaluación final sobre el conjunto de prueba.
79 print("\nEvaluando YOLOv5L + aumento de datos en el conjunto de prueba...")
80 !python val.py \
81     --weights runs/train/exp3_yolov5l_augment/weights/best.pt \
82     --data "/content/drive/My Drive/Yolo/config.yaml" \
83     --img 1024 \
84     --batch 8 \
85     --task test \
86     --name test_augment \
87     --exist-ok

```

Código A.1: Código del Experimento 3: YOLOv5L con aumento de datos avanzado

Apéndice B

Código de implementación del prototipo de detección en tiempo real

En el presente anexo se incluye el código utilizado para cargar el modelo entrenado, capturar imágenes mediante una cámara, realizar la inferencia, visualizar las detecciones y calcular la velocidad de procesamiento.

```
1 import argparse
2 import os
3 import pathlib
4 import platform
5 import time
6 import warnings
7
8 import cv2
9 import torch
10
11 warnings.filterwarnings(
12     "ignore",
13     message=r".*torch\.cuda\.amp\.autocast.*",
14     category=FutureWarning,
15 )
16
17
```

```

18 def parse_args():
19     parser = argparse.ArgumentParser(
20         description="Detección en tiempo real con YOLOv5"
21     )
22     parser.add_argument("--model", type=str, default="bestGanador.pt")
23     parser.add_argument("--source", type=str, default="0")
24     parser.add_argument("--conf", type=float, default=0.5)
25     parser.add_argument("--width", type=int, default=1280)
26     parser.add_argument("--height", type=int, default=720)
27     parser.add_argument("--infer-size", type=int, default=640)
28     parser.add_argument("--skip-frames", type=int, default=0)
29     parser.add_argument(
30         "--backend",
31         type=str,
32         default="auto",
33         choices=["auto", "dshow", "msmf"],
34     )
35     return parser.parse_args()
36
37
38 def main():
39     args = parse_args()
40
41     if not os.path.exists(args.model):
42         print(f"Error: no se encuentra el modelo en '{args.model}'")
43         return
44
45     # El parche solo se aplica en Windows y se restaura tras cargar el modelo.
46     original_posix_path = pathlib.PosixPath
47     if platform.system() == "Windows":
48         pathlib.PosixPath = pathlib.WindowsPath
49
50     try:
51         model = torch.hub.load(
52             "ultralytics/yolov5",
53             "custom",
54             path=args.model,
55             force_reload=False,

```

```

56     )
57     device = "cuda" if torch.cuda.is_available() else "cpu"
58     model.to(device)
59     model.conf = args.conf
60     print(f"Dispositivo de inferencia: {device}")
61 except Exception as exc:
62     print(f"Error al cargar el modelo: {exc}")
63     return
64 finally:
65     pathlib.PosixPath = original_posix_path
66
67 src = int(args.source) if args.source.isdigit() else args.source
68 backend_map = {"dshow": cv2.CAP_DSHOW, "msmf": cv2.CAP_MSMF}
69
70 if (
71     args.backend != "auto"
72     and platform.system() == "Windows"
73     and isinstance(src, int)
74 ):
75     cap = cv2.VideoCapture(src, backend_map[args.backend])
76 else:
77     cap = cv2.VideoCapture(src)
78
79 if not cap.isOpened():
80     print(f"Error: no se puede abrir la fuente {src}")
81     return
82
83 cap.set(cv2.CAP_PROP_FRAME_WIDTH, args.width)
84 cap.set(cv2.CAP_PROP_FRAME_HEIGHT, args.height)
85 actual_w = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
86 actual_h = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
87 print(
88     f"Resolución solicitada: {args.width}x{args.height} | "
89     f"resolución obtenida: {actual_w}x{actual_h}"
90 )
91
92 prev_time = time.time()
93 fps_history = []

```

```

94     frame_count = 0
95     last_detections = None
96     window_name = "YOLOv5 - Detección de imperfecciones"
97
98     cv2.namedWindow(window_name, cv2.WINDOW_NORMAL)
99     cv2.resizeWindow(window_name, actual_w, actual_h)
100
101     try:
102         while True:
103             ret, frame = cap.read()
104             if not ret:
105                 continue
106
107             frame_count += 1
108             run_inference = (
109                 args.skip_frames == 0
110                 or frame_count % (args.skip_frames + 1) == 0
111             )
112
113             if run_inference:
114                 results = model(frame, size=args.infer_size)
115                 last_detections = results.xyxy[0].cpu().numpy()
116
117             if last_detections is not None:
118                 for *box, conf, cls_id in last_detections:
119                     x1, y1, x2, y2 = map(int, box)
120                     label = f"{model.names[int(cls_id)]} {conf:.2f}"
121                     cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
122                     cv2.putText(
123                         frame,
124                         label,
125                         (x1, max(y1 - 10, 20)),
126                         cv2.FONT_HERSHEY_SIMPLEX,
127                         0.6,
128                         (0, 255, 0),
129                         2,
130                     )
131

```

```

132     curr_time = time.time()
133     instant_fps = 1 / (curr_time - prev_time)
134     prev_time = curr_time
135     fps_history.append(instant_fps)
136     if len(fps_history) > 15:
137         fps_history.pop(0)
138     avg_fps = sum(fps_history) / len(fps_history)
139
140     cv2.putText(
141         frame,
142         f"FPS: {avg_fps:.2f}",
143         (30, 60),
144         cv2.FONT_HERSHEY_SIMPLEX,
145         1,
146         (0, 255, 0),
147         2,
148     )
149     cv2.imshow(window_name, frame)
150
151     key = cv2.waitKey(1) & 0xFF
152     if key == ord("q") or key == 27:
153         break
154 except KeyboardInterrupt:
155     print("Programa detenido por el usuario")
156 finally:
157     cap.release()
158     cv2.destroyAllWindows()
159
160
161 if __name__ == "__main__":
162     main()

```

Código B.1: Código del prototipo de detección en tiempo real