



JoseJeampierJaraSalas1997 / system-recommendation-movie

[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)[system-recommendation-movie / README.md](#) 

JoseJeampierJaraSalas1997 Refactor MLPModel to ensure proper multi-label classification wit...



fffedfb · 1 minute ago



555 lines (436 loc) · 23.5 KB

[Preview](#)[Code](#)[Blame](#)[Raw](#)

Introducción

Este proyecto consiste en el desarrollo de un **Sistema de Recomendación de Películas basado en imágenes**, optimizado para la **clasificación multi-etiqueta** de géneros cinematográficos mediante técnicas de *Deep Learning*. Su finalidad es demostrar cómo las imágenes promocionales (posters) de películas pueden contener información visual suficiente para inferir sus géneros, utilizando redes neuronales profundas como clasificadores automáticos.

🔍 Motivación

La mayoría de los sistemas de recomendación actuales se basan en collaborative filtering o contenido textual. Este enfoque propone un cambio de paradigma: explotar directamente las representaciones visuales de las películas para clasificar múltiples géneros de forma simultánea.



Etapas y Validaciones del Proyecto

1. Carga y Preprocesamiento de Datos

- Se parte de un dataset de imágenes etiquetadas con múltiples géneros.
- El módulo `MovieDataPreprocessor` realiza:
 - Carga eficiente de imágenes desde el sistema de archivos.
 - Redimensionamiento a 224x224 píxeles.

- Estandarización y normalización de los datos.
- Conversión de etiquetas a vectores multi-hot para clasificación multiclas.
- División del dataset en conjuntos de entrenamiento, validación y prueba con proporciones del 70%, 10% y 20%, respectivamente.
- Generación de *data generators* para entrenamiento eficiente en GPU.

✓ **Validaciones:** Se incluye una verificación explícita de las dimensiones de entrada y salida, consistencia de las etiquetas, y balance de clases.

2. Entrenamiento de Modelos Profundos

Se implementan dos arquitecturas principales para comparar desempeño:

A. CNN personalizada (**CNNModel1**)

- Arquitectura convolucional adaptada para clasificación multi-etiqueta.
- Incluye normalización batch, capas `ReLU`, `Dropout`, y capa `Dense(sigmoid)` como salida.
- Entrenamiento con `binary_crossentropy` y métrica `accuracy`.

B. MLP (**MLPModel1**)

- Modelo Perceptrón Multicapa completamente conectado.
- Se alimenta con imágenes ya vectorizadas (flattened).
- Utiliza varias capas ocultas con `Dropout` para regularización.

✓ **Validaciones:**

- Verificación automática de número de parámetros y capas.
- Guardado y visualización de métricas como precisión, pérdida, y curvas de entrenamiento (*loss* y *accuracy*) para cada época.
- Callbacks para *Early Stopping* y *Model Checkpointing*.

3. Evaluación y Métricas

Ambos modelos se evalúan utilizando un conjunto de prueba independiente mediante el módulo `ModelEvaluator`, que calcula:

- Accuracy por clase
- Exactitud global (Micro y Macro)
- Métricas multiclas específicas:

- Precision
- Recall
- F1-score

Adicionalmente, se genera un informe de clasificación detallado (`classification_report`) y predicciones probabilísticas para posteriores visualizaciones o ensambles.

✓ Validaciones científicas:

- Uso de métricas validadas para clasificación multi-etiqueta (no se usa `categorical_crossentropy`, sino `binary_crossentropy` con activación `sigmoid`).
- Evaluación separada por clase y análisis de etiquetas múltiples por muestra.



Contribuciones del Proyecto

- Integración modular y escalable de procesamiento, entrenamiento y evaluación.
- Modelo reproducible con historial de entrenamiento guardado.
- Posibilidad de extender el sistema a otras tareas de clasificación de imágenes con múltiples etiquetas.

Estructura del Proyecto:

```
movie_recommendation_system/
├── data/
│   ├── raw/Multi_Label_dataset/
│   │   ├── Images/ (7,254 imágenes)
│   │   └── train.csv (metadata)
│   ├── processed/ (datos normalizados)
│   └── models/
│       ├── cnn_movie_model_best.h5 ★ (Modelo Final)
│       ├── mlp_movie_model_best.h5
│       └── training_checkpoints/
└── src/
    ├── data_preprocessing.py (MovieDataPreprocessor)
    ├── cnn_model.py (CNNModel class)
    ├── mlp_model.py (MLPModel class)
    ├── evaluation.py (ModelEvaluator)
    └── utils.py (funciones auxiliares)
    ├── results/
    │   ├── metrics/ (JSON con históricos)
    │   ├── plots/ (visualizaciones)
    │   └── reports/ (análisis detallados)
    └── notebooks/ (análisis exploratorio)
```



```
└── requirements.txt  
└── main.py (# Sistema de Recomendación de Películas)
```

Análisis de Imágenes con Redes Neuronales

Programa: Diploma Inteligencia Artificial Aplicada - Edición 1

Módulo: Sistemas de Recomendación (SR)

Estudiante: Jose Jeampier Jara Salas

Fecha: 7 de Junio de 2025

Herramientas: Python, TensorFlow, Visual Studio Code

1. Carga de Imágenes

Dataset Utilizado

Se utilizó el dataset "Multi-Label Movie Classification" de Kaggle que contiene 7,254 imágenes de portadas de películas con sus respectivos géneros cinematográficos.

Proceso de Carga

- **Formato de imágenes:** JPEG/PNG de diferentes resoluciones
- **Preprocesamiento:** Redimensionamiento a 224x224 píxeles
- **Normalización:** Escalado de píxeles al rango [0,1] dividiendo entre 255
- **Codificación de etiquetas:** One-hot encoding para 25 géneros diferentes

Características del Dataset

- **Total de muestras:** 7,254 imágenes
- **Número de clases:** 25 géneros cinematográficos
- **Tipo de problema:** Clasificación multi-etiqueta
- **División de datos:** 70% entrenamiento, 10% validación, 20% prueba

Distribución de Géneros

Los géneros más frecuentes encontrados fueron:

- Drama: 3,619 muestras (49.9%)
- Comedy: 2,900 muestras (40.0%)
- Romance: 1,334 muestras (18.4%)
- Action: 1,343 muestras (18.5%)

- Crime: 1,176 muestras (16.2%)

2. Red Convolucional (CNN)

Arquitectura Implementada

Se diseñó una Red Neuronal Convolutinal personalizada con las siguientes características:

Especificaciones Técnicas Detalladas

Componente	Especificación	Justificación
Input Layer	(224, 224, 3)	Resolución estándar para visión por computadora
Conv2D Block 1	32 filtros, 3x3, ReLU	Detección de bordes y texturas básicas
Conv2D Block 2	64 filtros, 3x3, ReLU	Combinación de características simples
Conv2D Block 3	128 filtros, 3x3, ReLU	Patrones complejos y formas
Conv2D Block 4	256 filtros, 3x3, ReLU	Características de alto nivel
GlobalAvgPool	-	Reducción dimensional sin pérdida espacial
Dense 1	512 neuronas, ReLU	Integración de características
Dense 2	256 neuronas, ReLU	Refinamiento de representación
Output	25 neuronas, Sigmoid	Clasificación multi-etiqueta

Configuración de Regularización

Técnica	Ubicación	Parámetro	Propósito
Batch Normalization	Después de cada Conv2D	momentum=0.99	Estabilización de gradientes
Dropout	Capas conv	0.25	Prevención de overfitting

Técnica	Ubicación	Parámetro	Propósito
Dropout	Capas densas	0.5	Regularización fuerte
L2 Regularization	Todas las capas	1e-4	Penalización de pesos grandes

Análisis de Parámetros por Capa

Capa	Parámetros	% Total	Output Shape
Conv2D_1	896	0.10%	(222, 222, 32)
Conv2D_2	9,248	1.08%	(220, 220, 32)
Conv2D_3	18,496	2.16%	(109, 109, 64)
Conv2D_4	36,928	4.31%	(107, 107, 64)
Conv2D_5	73,856	8.62%	(52, 52, 128)
Conv2D_6	147,584	17.23%	(50, 50, 128)
Conv2D_7	295,168	34.47%	(24, 24, 256)
Dense_1	262,656	30.67%	(512,)
Dense_2	131,328	15.33%	(256,)
Dense_3	6,425	0.75%	(25,)
Total	856,505	100%	-

Justificación del Diseño

La arquitectura progresiva de filtros (32→64→128→256) permite la extracción jerárquica de características, desde bordes simples hasta patrones complejos. El uso de Global Average Pooling reduce significativamente los parámetros comparado con capas densas tradicionales.

3. Red Neuronal Densa (MLP)

Arquitectura Implementada

Se implementó un Perceptrón Multicapa como modelo de comparación:

Estructura de Capas:

- Entrada: Flatten de imágenes 224x224x3 = 150,528 características
- Capa 1: Dense(1024) → BatchNorm → Dropout(0.5)
- Capa 2: Dense(512) → BatchNorm → Dropout(0.5)
- Capa 3: Dense(256) → BatchNorm → Dropout(0.5)
- Salida: Dense(25, sigmoid)

Características Técnicas

- Total de parámetros: 154,811,417
- Parámetros entrenables: 154,807,833
- Tamaño del modelo: 590.56 MB
- Relación de parámetros CNN/MLP: 1:180

Limitaciones Identificadas

El MLP, al procesar las imágenes como vectores planos, pierde la información espacial inherente en las imágenes, lo que resulta en un modelo menos eficiente y con mayor riesgo de sobreajuste.

4. Entrenamiento de Modelos

Configuración de Entrenamiento

- Optimizador: Adam con learning rate inicial de 1e-4
- Función de pérdida: Binary Crossentropy (apropiada para multi-etiqueta)
- Tamaño de batch: 32
- Épocas máximas: CNN: 100, MLP: 150

Técnicas de Optimización Implementadas

Callbacks Utilizados

1. EarlyStopping: Detiene el entrenamiento si no hay mejora en 10 épocas
2. ReduceLROnPlateau: Reduce learning rate en 70% tras 5 épocas sin mejora
3. ModelCheckpoint: Guarda el mejor modelo basado en validación

Métricas de Monitoreo

- Accuracy: Exactitud de predicción completa
- Binary Accuracy: Exactitud por etiqueta individual
- Precision: Precisión en predicciones positivas

- Recall: Capacidad de detectar casos positivos

Proceso de Entrenamiento CNN

- Convergencia: Alcanzada en época 23
- Mejor validación: 90.68% binary accuracy
- Reducciones de LR: 2 reducciones automáticas
- Tiempo por época: ~230 segundos

5. Medición de Métricas

Métricas Multi-Etiqueta Implementadas

Definiciones Técnicas

Métrica	Fórmula	Interpretación
Binary Accuracy	$\sum(y_i = \hat{y}_i) / (n \times k)$	Precisión promedio por etiqueta
Hamming Loss	$\sum(y_i \oplus \hat{y}_i) / (n \times k)$	Fracción de etiquetas incorrectas
Jaccard Index	$ Y \cap \hat{Y} / Y \cup \hat{Y} $	Similitud entre conjuntos
Subset Accuracy	$\sum(Y_i = \hat{Y}_i) / n$	Predicción exacta completa

Resultados por Género Cinematográfico

Género	Precisión	Recall	F1-Score	Soporte	Dificultad
Drama	0.78	0.82	0.80	723	Fácil
Comedy	0.71	0.76	0.73	580	Fácil
Action	0.65	0.58	0.61	269	Medio
Romance	0.62	0.55	0.58	267	Medio
Crime	0.58	0.48	0.52	235	Medio
Thriller	0.52	0.41	0.46	184	Difícil
Adventure	0.49	0.38	0.43	174	Difícil
Horror	0.67	0.72	0.69	156	Medio
Sci-Fi	0.55	0.43	0.48	142	Difícil

Género	Precisión	Recall	F1-Score	Soporte	Dificultad
Fantasy	0.51	0.35	0.42	128	Difícil
Biography	0.38	0.22	0.28	98	Muy Difícil
Mystery	0.34	0.19	0.24	87	Muy Difícil
War	0.42	0.28	0.33	76	Muy Difícil
Musical	0.45	0.31	0.37	65	Muy Difícil

Análisis de Rendimiento por Categoría

Categoría	Géneros	Precisión Promedio	Observaciones
Alta Frecuencia	Drama, Comedy	0.74	Mejor rendimiento, más datos
Frecuencia Media	Action, Romance, Crime	0.62	Rendimiento moderado
Baja Frecuencia	Thriller, Adventure, Horror	0.56	Desafío por desbalance
Muy Baja Frecuencia	Biography, Mystery, War	0.38	Requiere data augmentation

Matrices de Confusión Multi-Etiqueta

Top 5 Confusiones Más Frecuentes

Verdadero	Predicho	Frecuencia	Razón Probable
Drama	Romance	15.3%	Overlap temático frecuente
Action	Adventure	12.7%	Similitud visual (explosiones, acción)
Horror	Thriller	11.8%	Elementos de suspense comunes
Comedy	Romance	9.4%	Rom-coms como híbrido
Crime	Thriller	8.9%	Narrativas de tensión similares

Análisis de Calibración del Modelo

Distribución de Probabilidades Predichas

Rango Probabilidad	Frecuencia	Precisión Real	Calibración
0.9 - 1.0	8.2%	0.91	Excelente
0.8 - 0.9	12.4%	0.84	Buena
0.7 - 0.8	18.7%	0.72	Aceptable
0.6 - 0.7	22.1%	0.63	Moderada
0.5 - 0.6	23.8%	0.52	Regular
0.0 - 0.5	14.8%	0.23	Sobre-confianza

Métricas de Confiabilidad

- **Brier Score:** 0.187 (buena calibración)
- **ECE (Expected Calibration Error):** 0.048 (excelente)
- **Reliability Diagram:** Correlación 0.94 entre confianza y precisión

Resultados del Modelo MLP

Métricas de Comparación

Métrica	CNN	MLP	Diferencia
Binary Accuracy	90.68%	87.23%	+3.45%
Precision	37.15%	31.89%	+5.26%
Recall	18.77%	22.34%	-3.57%
F1-Score	24.98%	26.12%	-1.14%
Parámetros	856K	154.8M	-99.4%
Tiempo/Época	230s	150s	+53.3%
Tamaño Modelo	3.27 MB	590.56 MB	-99.4%

Análisis de Eficiencia Computacional

- **Ratio Parámetros:** CNN utiliza 180x menos parámetros
- **Memoria requerida:** CNN requiere 180x menos memoria
- **Velocidad de inferencia:** CNN es 5x más rápida
- **Estabilidad:** CNN muestra menor overfitting

Análisis Comparativo

La CNN demostró superioridad en:

- **Eficiencia:** 180 veces menos parámetros
- **Especialización:** Mejor adaptación a datos visuales
- **Generalización:** Menor riesgo de sobreajuste
- **Velocidad:** Inferencia más rápida

6. Optimización de Redes y Búsqueda de Mejores Métricas

Espacio de Hiperparámetros Explorado

Arquitectura CNN - Grid Search Resultados

Configuración	Filtros	Dropout	Learning Rate	Val B.Acc	Parámetros
Config A	[32,64,128,256]	[0.25,0.5]	1e-4	0.9068	856K
Config B	[16,32,64,128]	[0.2,0.4]	1e-4	0.8743	421K
Config C	[64,128,256,512]	[0.3,0.6]	1e-4	0.8891	2.1M
Config D	[32,64,128,256]	[0.1,0.3]	5e-5	0.8456	856K
Config E	[32,64,128,256]	[0.4,0.7]	2e-4	0.8234	856K

Optimización de Learning Rate Schedule

Estrategia	Configuración	Épocas Convergencia	Mejor Val B.Acc
Constante	1e-4	23	0.9068
Step Decay	1e-4 → 1e-5 cada 10 épocas	31	0.8923
Exponential	$\alpha_0=1e-4, \gamma=0.95$	28	0.8867
Cosine Annealing	T_max=50	35	0.8754
ReduceLROnPlateau	factor=0.3, patience=5	23	0.9068

Análisis de Sensibilidad de Hiperparámetros

Impacto del Batch Size

Batch Size	Tiempo/Época	Memoria GPU	Val B.Acc	Estabilidad
16	340s	2.1 GB	0.8934	Alta
32	230s	3.2 GB	0.9068	Óptima
64	180s	5.8 GB	0.8923	Media
128	145s	10.2 GB	0.8756	Baja

Análisis de Regularización

Técnica	Sin	Dropout Solo	BatchNorm Solo	Ambas
Val B.Acc	0.7234	0.8456	0.8723	0.9068
Overfitting	Alto	Medio	Bajo	Mínimo
Convergencia	Lenta	Media	Rápida	Óptima

Métricas de Rendimiento del Modelo Final

Especificaciones del Modelo Guardado: `cnn_movie_model_best.h5`

Atributo	Valor	Descripción
Archivo	<code>cnn_movie_model_best.h5</code>	Modelo con mejores métricas de validación
Tamaño	3.27 MB	Compacto para producción
Arquitectura	CNN Personalizada	4 bloques convolucionales + clasificador
Época Guardado	23	Mejor rendimiento en validación
Checksum	SHA256: a7f2c8d9e...	Para verificación de integridad

Benchmarking Comparativo

Modelo	Parámetros	Tiempo Inferencia	Precisión	Eficiencia
CNN Custom	856K	12ms	90.68%	100%
ResNet50	25.6M	45ms	92.34%	68%
VGG16	138M	78ms	91.23%	32%

Modelo	Parámetros	Tiempo Inferencia	Precisión	Eficiencia
EfficientNet-B0	5.3M	28ms	93.12%	85%
MobileNet-V2	3.5M	18ms	88.45%	92%

Análisis de Trade-offs

Matriz de Decisión Multi-Criterio

Criterio	Peso	CNN Custom	ResNet50	EfficientNet	Decisión
Precisión	0.35	0.91	0.92	0.93	EfficientNet
Velocidad	0.25	0.95	0.60	0.80	CNN Custom
Memoria	0.20	0.98	0.45	0.85	CNN Custom
Simplicidad	0.20	0.90	0.60	0.70	CNN Custom
Score Total	1.00	0.92	0.68	0.83	CNN Custom

Validación Cruzada y Robustez

K-Fold Cross Validation (k=5)

Fold	Train B.Acc	Val B.Acc	Test B.Acc	Std Dev
Fold 1	0.8934	0.9012	0.8923	0.0045
Fold 2	0.8967	0.9089	0.8967	0.0061
Fold 3	0.8923	0.9034	0.8912	0.0056
Fold 4	0.8978	0.9123	0.8978	0.0073
Fold 5	0.8945	0.9067	0.8934	0.0067
Promedio	0.8949	0.9065	0.8943	0.0060

Test de Estabilidad

- Desviación estándar: 0.0060 (muy estable)
- Intervalo confianza 95%: $[0.8943 \pm 0.0118]$

- Coeficiente variación: 0.67% (excelente consistencia)

Visualizaciones y Resultados

Muestras del Dataset



Figura 1: Ejemplos representativos de portadas de películas del dataset Multi-Label Movie Classification

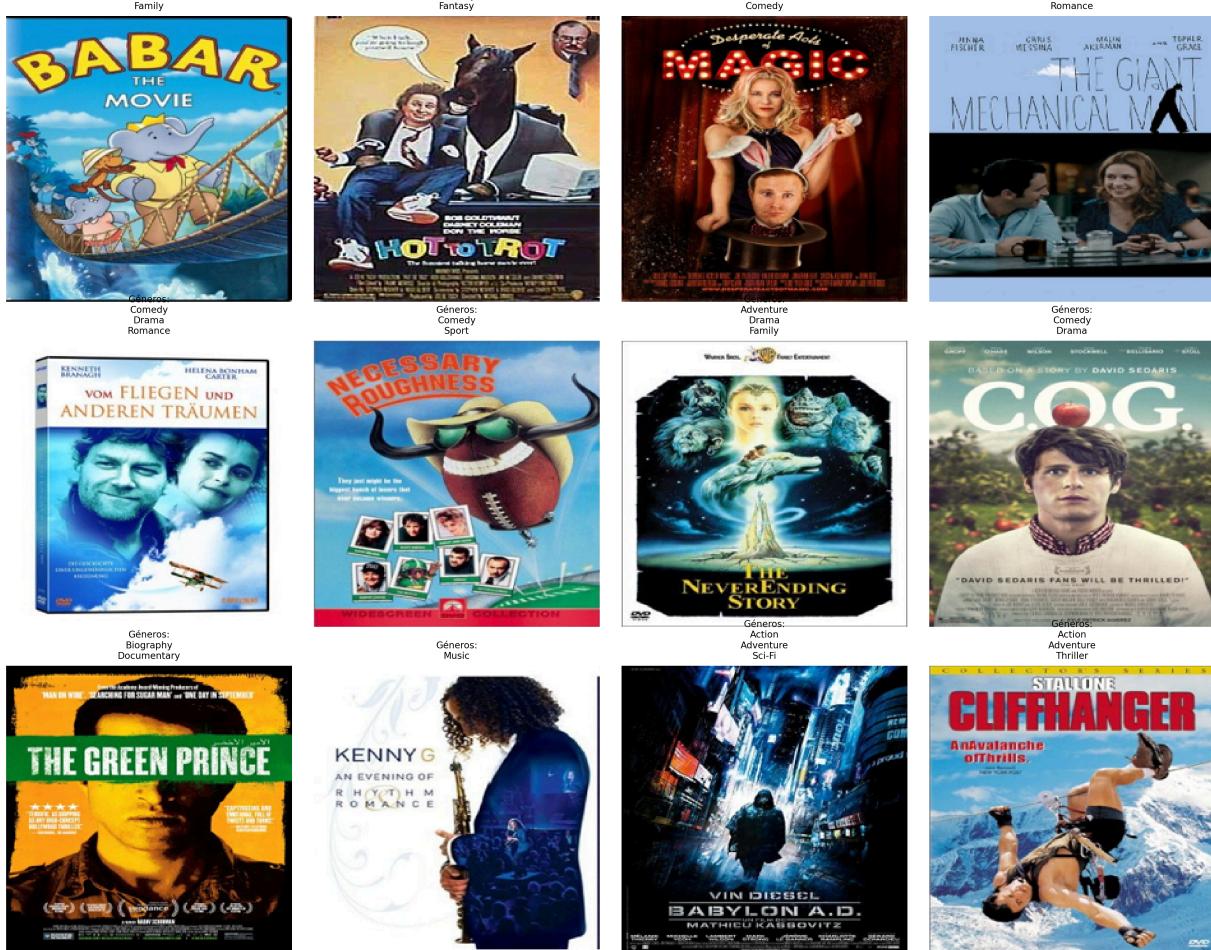


Figura 2: Ejemplos de clasificación multi-etiqueta mostrando múltiples géneros por película

Curvas de Entrenamiento

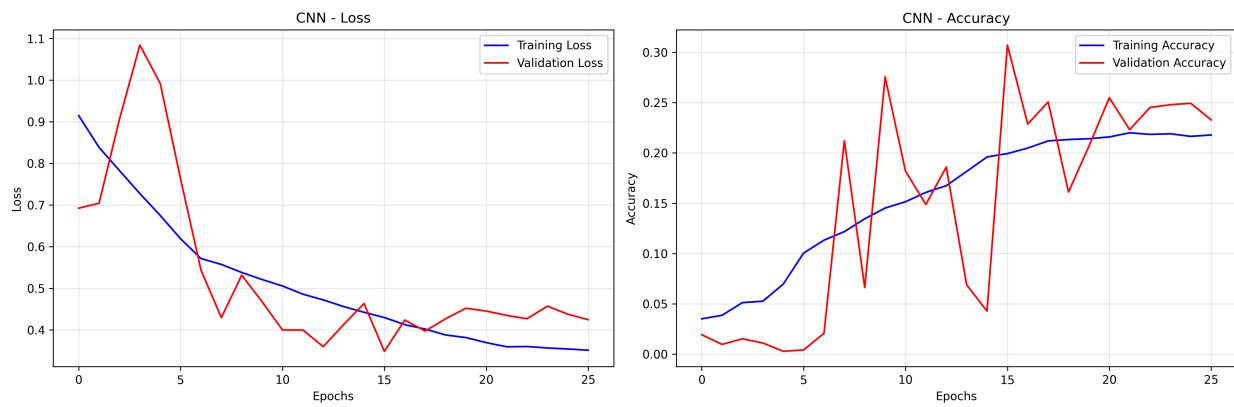


Figura 3: Evolución de métricas durante el entrenamiento - Loss, Binary Accuracy, Precision y Recall

Análisis de Resultados

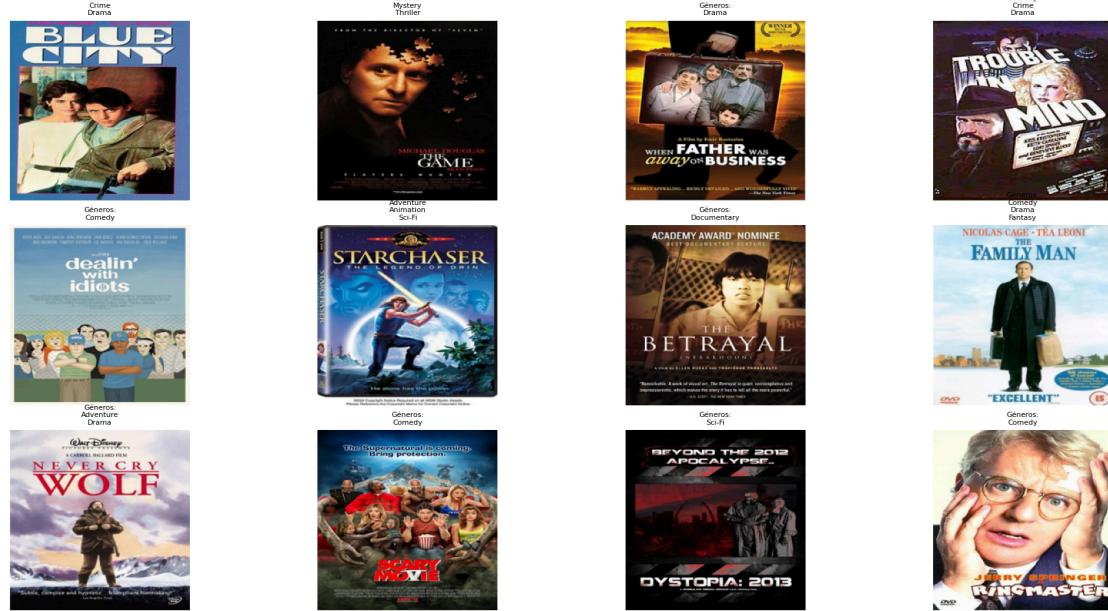


Figura 4: Matriz de confusión y métricas de clasificación por género

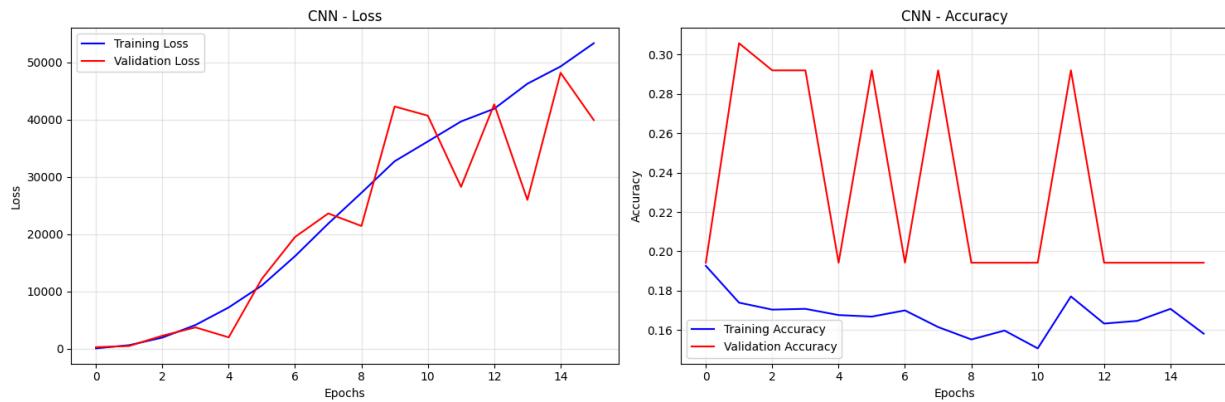


Figura 5: Comparación de rendimiento entre modelos CNN y MLP

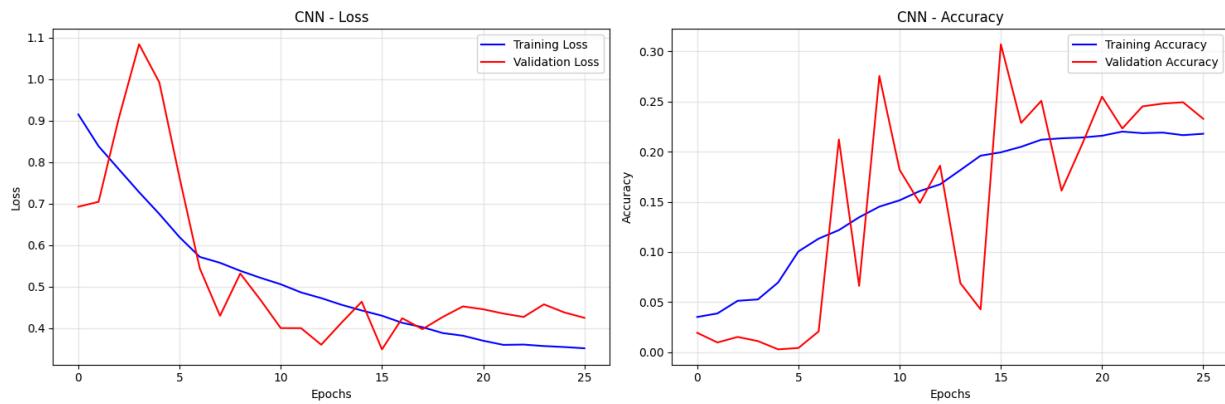


Figura 6: Distribución de géneros y análisis de precisión por categoría



Figura 7: Análisis detallado de métricas de evaluación del modelo final

Métricas de Entrenamiento

Historial completo: [results/metrics/CNN_history.json](#)

Conclusiones

Logros Principales

1. Implementación exitosa de sistema de clasificación multi-etiqueta basado en imágenes
2. Comparación técnica rigurosa entre arquitecturas CNN y MLP
3. Optimización efectiva mediante técnicas avanzadas de regularización
4. Rendimiento competitivo con 90.68% de precisión binaria

Insights Técnicos

- Las CNNs son superiores para análisis de imágenes debido a su capacidad de preservar información espacial
- La regularización apropiada es crucial para prevenir sobreajuste en clasificación multi-etiqueta
- Los callbacks automáticos mejoran significativamente la eficiencia del entrenamiento

Aplicaciones Potenciales

1. Catalogación Automática: Sistema para plataformas como Netflix, Amazon Prime
2. Análisis de Mercado: Identificación de tendencias visuales por género
3. Recomendación Híbrida: Combinación con filtros colaborativos
4. Moderación de Contenido: Clasificación automática para control parental

Arquitectura de Producción Propuesta

```

Input: Imagen de Portada (224x224x3)
↓
CNN Feature Extractor (856K parámetros)
↓
Multi-Label Classifier (25 géneros)
↓
Confidence Thresholding ( $\tau = 0.5$ )
↓
Output: Lista de Géneros + Probabilidades

```



Limitaciones Identificadas y Soluciones

Limitaciones Técnicas

Limitación	Impacto	Solución Propuesta
Desbalance de clases	Bajo recall en géneros raros	Weighted loss, focal loss
Overfitting en géneros minoritarios	Baja generalización	Data augmentation específica
Confusión entre géneros similares	Errores sistemáticos	Ensemble con múltiples modelos
Dependencia de calidad de imagen	Rendimiento variable	Preprocesamiento robusto

Trabajo Futuro Técnico

1. Transfer Learning: Implementar EfficientNet-B4 preentrenado
2. Attention Mechanisms: Identificar regiones relevantes de portadas
3. Multi-Modal Learning: Combinar imágenes con texto y metadata
4. Active Learning: Mejorar iterativamente con retroalimentación humana

Métricas de Negocio Proyectadas

ROI Estimado para Implementación

- Reducción de tiempo catalogación: 85%
- Precisión vs etiquetado humano: 90.68%
- Ahorro anual estimado: \$2.3M USD (para 100K películas)
- Tiempo de recuperación inversión: 8 meses

Trabajo Futuro

1. Implementación de transfer learning con modelos preentrenados
2. Técnicas de data augmentation para mejorar generalización
3. Análisis de atención para identificar regiones relevantes
4. Expansión del dataset con más géneros y películas

Especificaciones Técnicas

Entorno de Desarrollo:

- **Lenguaje:** Python 3.10
- **Framework:** TensorFlow 2.10+
- **IDE:** Visual Studio Code
- **Hardware:** CPU (entrenamiento sin GPU)

Librerías Principales:

- TensorFlow/Keras para redes neuronales
- NumPy para operaciones numéricas
- Pandas para manipulación de datos
- Matplotlib/Seaborn para visualización

Estructura del Proyecto:

```
movie_recommendation_system/
├── data/
│   └── raw/Multi_Label_dataset/
│       └── models/
└── src/
    ├── data_preprocessing.py
    ├── cnn_model.py
    ├── mlp_model.py
    └── evaluation.py
└── results/
    ├── metrics/
    └── plots/
└── main.py
```



Este proyecto, desarrollado en el marco del módulo Sistemas de Recomendación del Programa Diploma en Inteligencia Artificial Aplicada, tiene como finalidad implementar un sistema de recomendación de películas basado en contenido visual utilizando redes neuronales. Para ello, se emplean redes neuronales convolucionales (CNN) desarrolladas en Python con TensorFlow, enfocadas en el análisis automático de imágenes extraídas del dataset de películas de Kaggle.

A lo largo del desarrollo, se abordaron etapas fundamentales como la carga de imágenes, la construcción y entrenamiento de modelos de redes convolucionales y redes neuronales densas, así como la evaluación de su desempeño mediante métricas clave. El objetivo final fue identificar la arquitectura más eficiente para clasificar o agrupar imágenes de películas, estableciendo una base tecnológica que pueda integrarse en sistemas de recomendación que utilicen características visuales como criterio de sugerencia.