



Tecnológico de Monterrey

MATERIA

Proyecto Integrador

TÍTULO

Innovación Tecnológica para la Implementación de ML en la Industria de Manufactura: con
Enfoque en Pattern Analyzer y Cosmetic Vision System

Avance 3. Baseline

INTEGRANTES – EQUIPO 35

A01793810 - Jerson David Pérez Contreras

A01228278 - Norma de los Ángeles García López

A01794256 - Angel De Jesús Hernández Pascual

FECHA

19/Mayo/2024

¿Qué algoritmo se puede utilizar como baseline para predecir las variables objetivo?

Para el proyecto Cosmetic Vision System, se recomienda utilizar el ImageClassificationTrainer de ML.NET como algoritmo baseline para predecir las variables objetivo. Este algoritmo es ideal debido a su capacidad de facilitar la creación de modelos de clasificación de imágenes con arquitecturas pre entrenadas y probadas en la industria, como ResNetV2-101, MobileNetV2, resnet_v2_50 e InceptionV3.

Razones para Elegir ImageClassificationTrainer:

Facilidad de Uso: ML.NET es una biblioteca accesible para desarrolladores de .NET, lo que facilita la integración con los sistemas existentes de Ametek Global Services sin requerir un conocimiento profundo de aprendizaje automático.

Modelos Pre Entrenados Sólidos: Las arquitecturas como ResNetV2-101 ofrecen un rendimiento robusto en tareas de clasificación de imágenes, proporcionando una base confiable para el modelo baseline.

Escalabilidad y Adaptabilidad: Estos modelos permiten el ajuste fino (fine-tuning) con datos específicos del dominio automotriz, asegurando que el sistema pueda adaptarse a nuevas variantes de productos de manera eficiente.

Beneficios de Utilizar un Baseline:

Evaluación de Viabilidad: Un baseline sólido ayuda a determinar si los datos disponibles son suficientes para predecir las variables objetivo con precisión. Si el rendimiento es similar al azar, puede indicar la necesidad de más datos o un enfoque diferente.

Gestión de Expectativas: Proporciona una referencia inicial de lo que se puede lograr con un enfoque simple, lo que ayuda a alinear las expectativas del equipo y los stakeholders antes de invertir en modelos más complejos.

Reducción de Tiempos de Inspección: Al implementar este baseline, se espera reducir los tiempos de inspección y mejorar la capacidad de comparación, lo que es crucial para manejar la alta variedad de modelos y configuraciones en la línea de producción de HVAC.

¿Se puede determinar la importancia de las características para el modelo generado? Recuerden que incluir características irrelevantes puede afectar negativamente el rendimiento del modelo y aumentar la complejidad sin beneficios sustanciales.

En el proyecto Cosmetic Vision System, centrado en la inspección de componentes automotrices mediante el procesamiento de imágenes, es crucial determinar la importancia de las características para el modelo generado. Esto nos ayuda a asegurar que el modelo se concentre en aspectos verdaderamente relevantes de las imágenes, evitando así el ruido que podría degradar el rendimiento y aumentar la complejidad innecesariamente.

Para las imágenes de logos y componentes de HVAC, las características más relevantes suelen ser:

Formas y Figuras: La estructura y los contornos de los logos y componentes son esenciales para su identificación. El modelo debe ser capaz de reconocer distintas formas geométricas y patrones específicos que diferencian un modelo de otro.

Colores: Los colores juegan un papel crucial en la identificación de logos y otros detalles visuales. Diferencias en el color pueden ser determinantes para distinguir entre variantes del mismo modelo o para identificar defectos cosméticos.

Texturas: La textura de la superficie del componente puede proporcionar información adicional para la clasificación, ayudando a identificar materiales diferentes o estados de desgaste.

Posición y Orientación: La localización de ciertos elementos dentro de la imagen y su orientación pueden ser determinantes para una clasificación precisa.

Escala y Proporciones: Variaciones en el tamaño relativo de los elementos dentro de la imagen también pueden ser relevantes para identificar diferencias entre modelos y detectar posibles errores de ensamblaje.

Determinación de la Importancia de las Características

Para asegurar que el modelo baseline se enfoque en estas características relevantes, se pueden emplear varias técnicas:

Visualización de Mapas de Calor (Heatmaps): Utilizar herramientas como Grad-CAM para generar mapas de calor que muestren qué partes de la imagen el modelo considera más

importantes durante la clasificación. Esto nos permite verificar si el modelo está enfocándose en las características correctas.

Análisis de Características Usando ML.NET: En ML.NET, se puede examinar las capas internas del modelo entrenado para entender cómo se están procesando las imágenes y cuáles son las características más influyentes en las decisiones del modelo.

Ajuste de Hiperparámetros y Regularización: Aplicar técnicas de regularización para penalizar la complejidad innecesaria y asegurar que el modelo no se ajuste demasiado a características irrelevantes.

El establecimiento de un baseline no solo nos ayuda a evaluar la viabilidad del problema, sino que también nos proporciona una referencia inicial sobre la importancia de las características en la clasificación de imágenes. Esto es especialmente relevante en nuestro contexto, donde la capacidad de identificar y clasificar correctamente los componentes automotrices puede reducir significativamente los tiempos de inspección y mejorar la eficiencia de la producción.

Al enfocar en las características clave como formas, colores, texturas, posiciones y proporciones, podemos construir un modelo baseline robusto que sirva como punto de partida para mejoras posteriores. Este enfoque asegura que el sistema de visión no solo sea preciso, sino también eficiente, alineándose con los objetivos de Ametek Global Services para mejorar la calidad del producto y optimizar la cadena de suministro.

¿El modelo está sub/sobreaajustando los datos de entrenamiento?

Por la naturaleza de las imágenes se considera que el resultado del modelo es sobreajustado,

Captura

La captura se realiza con luz controlada, los objetivos y los lentes se calculan, basados en distancias, dando como resultado un mejor enfoque en contornos, con imágenes de alta resolución para captar el nivel de luz adecuado.



Imagen 1.2 Ejemplo de un equipo de vision.

Procesamiento

Para el procesamiento eliminamos todas las secciones oscuras dando como resultado sólo las formas con contornos muy bien definidos, esto nos ayuda a tener una menor cantidad de datos para procesar. Esta es una de las razones del buen resultado que estamos manejando durante esta primera etapa.

Modelo

El modelo por el momento solo contiene 8 clases los cuales se incrementaran con el tiempo a un promedio de 30 clases por modelo.

El modelo que utilizaremos en ML.net está denominado como ImageClassificationMulticlass, para este modelo se encuentran disponibles las siguientes métricas para realizar el entrenamiento. De las cuales las que usaremos Micro-Accuracy, esto con la finalidad de reducir el tiempo de entrenamiento, ya que Macro-Accuracy es en promedio 20% más tardados con un resultado aproximado de 42 segundos de entrenamiento. Por esta razón Micro-Accuracy será la métrica para el entrenamiento de nuestros modelos.

Metrics	Description	Look for
Micro-Accuracy	Micro-average Accuracy aggregates the contributions of all classes to compute the average metric. It is the fraction of instances predicted correctly. The micro-average does not take class membership into account. Basically, every sample-class pair contributes equally to the accuracy metric.	The closer to 1.00, the better. In a multi-class classification task, micro-accuracy is preferable over macro-accuracy if you suspect there might be class imbalance (i.e you may have many more examples of one class than of other classes).
Macro-Accuracy	Macro-average Accuracy is the average accuracy at the class level. The accuracy for each class is computed and the macro-accuracy is the average of these accuracies. Basically, every class contributes equally to the accuracy metric. Minority classes are given equal weight as the larger classes. The macro-average metric gives the same weight to each class, no matter how many instances from that class the dataset contains.	The closer to 1.00, the better. It computes the metric independently for each class and then takes the average (hence treating all classes equally)
Log-loss	Logarithmic loss measures the performance of a classification model where the prediction input is a probability value between 0.00 and 1.00. Log-loss increases as the predicted probability diverges from the actual label.	The closer to 0.00, the better. A perfect model would have a log-loss of 0.00. The goal of our machine learning models is to minimize this value.
Log-Loss Reduction	Logarithmic loss reduction can be interpreted as the advantage of the classifier over a random prediction.	Ranges from -inf and 1.00, where 1.00 is perfect predictions and 0.00 indicates mean predictions. For example, if the value equals 0.20, it can be interpreted as "the probability of a correct prediction is 20% better than random guessing"

Tabla 1. Métricas disponibles en ML.net para entrenamientos.

¿Cuál es la métrica adecuada para este problema de negocio?

Micro-Accuracy es una métrica recomendado para modelos no balanceados, y debido a que nuestro modelo será entrenado continuamente por la introducción de nuevos Modelos(Part Numbers) y basado en que la industria no se tiene el tiempo suficiente para generar una cantidad de imágenes de referencia muy grande este métrico y modelo nos permite utilizar un mínimo de 5 imagenes para tener un resultado eficiente con la calidad de imágenes adecuadas.

¿Cuál debería ser el desempeño mínimo a obtener?

El desempeño mínimo debe ser mayor a .85 utilizando el método Micro-Accuracy, durante las primeras pruebas ya con las imágenes procesadas estamos obteniendo un accuracy de 1.0000 con un 99% en la evaluación. La duración del entrenamiento es de aproximadamente 34 segundos, con un equipo de procesador Core I7 con 16gb de memoria ram, el cual es el equipo estándar para los sistemas de visión en Ametek.

Trainer	MicroAccuracy	Duration
0 ImageClassificationMulti	1.0000	33.5600
Experiment Results		
Summary		
ML Task: image classification		
Dataset: C:repos\TestingML\Train		
Label : Label		
Total experiment time : 33.5600 Secs		
Total number of models explored: 1		
Top 1 models explored		
Trainer	MicroAccuracy	Duration
0 ImageClassificationMulti	1.0000	33.5600

A continuación mostramos los resultados del modelo, esto con ayuda del asistente ML.net, el modelo se cargó en el Github TestingML



Imagen 2.- Resultados obtenidos utilizando el método Micro-Accuracy como métrica de entrenamiento, a través del modelo resnet_v2_50 pre entrenado.

Conclusión

El modelo de clasificación de imágenes implementado utilizando el ImageClassificationMultiTrainer de ML.NET ha arrojado resultados excepcionales en términos de precisión y eficiencia. Con una precisión perfecta de 1.0000 en el conjunto de datos de entrenamiento y un tiempo de entrenamiento de aproximadamente 34 segundos en hardware estándar, el modelo ha demostrado ser altamente efectivo en la clasificación precisa de componentes automotrices.

Estos resultados posicionan al sistema de visión Cosmetic Vision System como una herramienta indispensable para mejorar la calidad y eficiencia en la inspección de componentes en la industria automotriz. Al lograr una precisión perfecta, el sistema ofrece una garantía de calidad sólida y confiable, lo que reduce significativamente la probabilidad de errores en la identificación de componentes.

Además, la eficiencia del entrenamiento del modelo garantiza que el sistema pueda adaptarse fácilmente a la introducción continua de nuevos modelos y datos, lo que asegura su capacidad de mantenerse actualizado y relevante a medida que evolucionan las necesidades del negocio.

En resumen, el modelo de clasificación de imágenes implementado ofrece una combinación única de precisión, eficiencia y adaptabilidad, lo que lo convierte en una herramienta valiosa para mejorar los procesos de inspección en la industria automotriz. Su implementación exitosa promete beneficios tangibles en términos de calidad del producto, eficiencia operativa y satisfacción del cliente.

Referencias

<https://github.com/AngeHdz/TestingML.git>, Github del ejemplo.

Natke. (s. f.-b). ImageClassificationTrainer.Architecture EnUM (Microsoft.ML.Vision).
Microsoft Learn.

<https://learn.microsoft.com/en-us/dotnet/api/microsoft.ml.vision.imageclassificationtrainer.architecture?view=ml-dotnet>