Software de procesamiento automático de placas espectrográficas

N. Pereyra[1,2], S. Ponte Ahon[2], Y.J. Aidelman[1,3], F. Ronchetti[2,4], F. Quiroga[2], R. Gamen[1,3] & L. Cidale[1,3]

[1] *Facultad de Ciencias Astronómicas y Geofísicas, UNLP, Argentina*
[2] *Instituto de Investigación en Informática, Facultad de Informática, UNLP, Argentina*
[3] *Instituto de Astrofísica de La Plata, CONICET–UNLP, Argentina*
[4] *Comisión de Investigaciones Científicas de la Provincia de Buenos Aires, Argentina*

Contacto / aidelman@fcaglp.unlp.edu.ar

Resumen / La Facultad de Ciencias Astronómicas y Geofísicas, a través del proyecto de Recuperación del Trabajo Observacional Histórico (ReTrOH), se encuentra realizando un proceso de digitalización de una gran colección de placas espectroscópicas en formato de vidrio. Por otro lado, en la actualidad las Redes Neuronales son los modelos de Aprendizaje Automático con mejor desempeño capaces de resolver una gran variedad de problemas. Son modelos generales y aproximadores universales. En los últimos años, se ha conseguido entrenar Redes Neuronales con múltiples capas mediante un conjunto de técnicas que suelen denominarse Aprendizaje Profundo. En este contexto, estamos desarrollando un software de procesamiento automático de las placas espectrográficas, que detecta los espectros de ciencia individuales que en estas hubiera con Aprendizaje Profundo y permite, además, cargar sus respectivos metadatos.

Abstract / The Faculty of Astronomical and Geophysical Sciences, through the Recovery of Historical Observational Work (ReTrOH, by its acronym in Spanish) project, is in the process of digitizing a large collection of glass-format spectroscopic plates. On the other hand, Neural Networks are currently the best performing Machine Learning models capable of solving a wide variety of problems. They are general models and universal approximators. In recent years, multi-layered Neural Networks have been successfully trained using a set of techniques often referred to as Deep Learning. In this context, we are developing a software for automatic processing of spectrographic plates, which detects the individual science spectra on these plates using Deep Learning and allows, in addition, to load their respective metadata.

Keywords / astronomical databases: miscellaneous — virtual observatory tools

1. Marco teórico

En la actualidad, el procesamiento de imágenes ha obtenido una gran importancia en diferentes áreas tales como la medicina, las telecomunicaciones, el entretenimiento, la astronomía, entre otros. Principalmente, esto se debe a las múltiples posibilidades de manipulación que ofrece para adquirir información de dichas imágenes. Por lo tanto, este tipo de proceso consiste en un conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información. Hoy en día, las Redes Neuronales son los modelos de aprendizaje automático con mejor desempeño capaces de resolver una gran variedad de problemas. Son modelos generales y aproximadores universales. En los últimos años, se ha conseguido entrenar Redes Neuronales con múltiples capas mediante un conjunto de técnicas que suelen denominarse Aprendizaje Profundo (*Deep Learning*).

En este contexto, la Facultad de Ciencias Astronómicas y Geofísicas de la UNLP, a través del proyecto ReTrOH (Recuperación del Trabajo Observacional Histórico)*, está realizando un proceso de digitalización de una gran colección de placas espectroscópicas en formato de vidrio. De este modo, es posible el almacenamiento perpetuo de información adquirida durante casi un siglo y su posterior procesamiento (Meilán, 2018; Meilán et al., 2020, 2022).

El almacenamiento adecuado de los espectros digitalizados es crucial para que tengan valor científico y puedan utilizarse en futuros trabajos. Por este motivo, es necesario que cada espectro escaneado se guarde en un archivo con formato FITS (*Flexible Image Transport System*) con su correspondiente encabezado (*header*) en el cual se vuelque toda la información que describe las observaciones. Este proceso debe realizarse para cada espectro ya que las características de la observación (como la hora y la fecha) son diferentes para cada uno. Además, en la mayoría de las placas hay registrados más de un espectro, por lo que es necesario segmentarlos de la imagen de la placa escaneada.

Para resolver estas problemáticas, comenzamos el desarrollo de un software cuya funcionalidad inicial comprende: (1) identificar de manera automática los espectros que se encuentran en la imagen; (2) definir regiones rectangulares que contengan a cada espectro; (3) cargar los metadatos con la información que quedará registra-

*https://retroh.fcaglp.unlp.edu.ar/

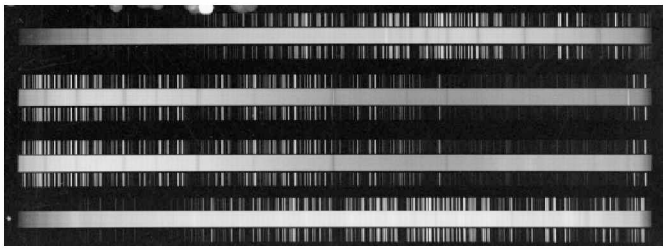

Figura 1: Ejemplo de una placa espectrográfica digitalizada y utilizada para aplicar técnicas de aprendizaje profundo.

da en el encabezado del archivo FITS; (4) guardar en formato FITS cada uno de los espectros.

2. Análisis de las placas espectrográficas

Se analizaron dos colecciones digitalizadas de placas espectroscópicas. La primera, 111 placas, corresponde a una parte de la colección de Virpi Niemela (VN), las cuales están disponibles en el repositorio institucional de la UNLP, SEDICI** (Servicio de Difusión de la Creación Intelectual). La segunda colección, fue digitalizada en el ICATE (Instituto de Ciencias Astronómicas, de la Tierra y el Espacio, UNSJ-CONICET) y consta de 154 imágenes formato TIFF (*Tagged Image File Format*).

Para trabajar con el formato FITS se utilizó el lenguaje Python y en específico la librería Astropy (Astropy Collaboration et al., 2013, 2018, 2022). El proyecto de Astropy tiene como objetivo desarrollar una librería común para resolver problemas del área de astronomía en Python. Además se utilizó Jupyter Notebooks (Kluyver et al., 2016) como interfaz para visualizar los datos. En la Fig. 1 se puede observar una placa digitalizada, la cual posee 4 espectros de ciencia en el centro y las lámparas de comparación correspondientes a sus lados.

2.1. Recorte de los espectros de forma automática

En una segunda etapa se automatizó el recorte e individualización de cada espectro registrado en las placas. Para ello se utilizó un modelo de detección de objetos llamado YOLO (YOLOv1 Redmon et al., 2016) en su versión 5 (YOLOv5 Jocher, 2022). YOLO permite realizar la detección de objetos arbitrarios con técnicas de aprendizaje profundo. Para ello, se debe entrenar el modelo con imágenes de los objetos de interés etiquetadas. La descripción de arquitectura de la red completa está fuera del alcance de este artículo, pero se describe en detalle en el repositorio del mismo***.

Para re-entrenar el modelo YOLO se utilizaron las imágenes de las placas, pero reduciendo tanto la profundidad de color como las dimensiones para acelerar el proceso de entrenamiento.

El proceso de recorte automático de los espectros, se subdivide en tres etapas: el etiquetado, el entrenamiento del modelo y el recorte automático. A continuación se detallan cada una de ellas.

Figura 2: Etiquetado de una placa espectroscópica digitalizada.

Proceso de etiquetado: El etiquetado de datos consiste en identificar datos sin procesar (imágenes, archivos de texto, videos, etc.) y agregar una o más etiquetas significativas e informativas para proporcionar un contexto, para que luego un modelo de aprendizaje profundo pueda aprender de ellos. Para realizar el trabajo de etiquetado, se utilizó el software Label Studio**** el cual es de código abierto y permite exportar a múltiples formatos los datos etiquetados (incluido el formato que utiliza YOLOv5). Con este software, se etiquetaron las 111 placas espectroscópicas correspondientes a una porción de la colección VN y las 154 de la colección ICATE. En la Fig. 2 se observa una placa espectroscópica etiquetada con el software mencionado, donde los recuadros verdes corresponden al etiquetado.

Entrenamiento del modelo: Para llevar a cabo el entrenamiento del modelo, se separó el conjunto de datos en dos partes: el conjunto de entrenamiento, el cual se utiliza para que el modelo aprenda a realizar la tarea requerida, y un conjunto de evaluación, el cual se utiliza para verificar cómo funciona el modelo. Para que el modelo tenga más datos para entrenar y pueda aprender a identificar los espectros de mejor manera, se utilizó una técnica llamada aumentación de datos (*data augmentation*) para las imágenes de entrenamiento. Esta técnica consiste en transformar de forma aleatoria pero leve todas las imágenes de entrenamiento del modelo, permitiendo aumentar su variabilidad. YOLOv5 tiene esta técnica integrada en el propio entorno de trabajo y fue utilizado para configurar las transformaciones. Para realizar el entrenamiento, se utilizaron las siguientes configuraciones:

- Se utilizó la versión *nano* de YOLOv5, que tiene un

**http://sedici.unlp.edu.ar/handle/10915/74497

***Repositorio YOLOv5

****https://labelstud.io/

bajo consumo de memoria y cómputo.

- Se estableció un 90 % del conjunto de datos para el conjunto de entrenamiento y un 10 % para el conjunto de evaluación; no fue necesaria una búsqueda de hiperparámetros. El tamaño de imagen de entrada es de 512×512 pixeles.
- Se aplicaron las siguientes transformaciones utilizando *data augmentation*: un cambio de brillo del 30 % como máximo, ruido gaussiano en un rango del 10 % al 50 %, rotación completa de la imagen de forma vertical y otra de máximo 3 grados, un escalado de máximo 20 % y la técnica de mosaico la cual permite que el modelo aprenda a identificar a los objetos a una escala más pequeña de lo normal. Se verificó visualmente que dichas transformaciones se encuentran dentro del rango de lo esperable en el dominio original.
- Se utilizó un tamaño de imagen de 512 pixeles.
- Se entrenó el modelo en un periodo de 250 épocas con lotes de 64 ejemplos. El optimizador utilizado fue AdamW, con una tasa de aprendizaje inicial de 0.01, y una variación cíclica de la tasa de aprendizaje.

Para analizar correctamente el aprendizaje del modelo, es necesario observar las métricas de precisión (*precision*) y sensibilidad (*recall*). La precisión indica cuántos ítems reconocidos son realmente relevantes. La sensibilidad, por otro lado, nos indica cuántos ítems relevantes fueron realmente seleccionados. El caso ideal sería tener una precisión de uno y una sensibilidad de uno, pero es difícil de lograr, en general al aumentar uno el otro disminuye. Una vez finalizado el entrenamiento de la Red Neuronal el modelo alcanza un 0.99 de precisión y de sensibilidad en el conjunto de entrenamiento. Por lo tanto, ha logrado generalizar y detectar los espectros en las imágenes.

Segmentación automática: En esta última etapa, se realiza la segmentación de forma automática de los espectros detectados en una imagen. Para ello, primero se transforma la imagen de la placa escaneada a un tamaño de 512×512 píxeles y se la utiliza como entrada del modelo entrenado. Como salida se obtienen las cajas de las predicciones, donde cada caja corresponde a un espectro detectado. Las cajas están conformadas por los cuatro vértices del área detectada del espectro. Una vez detectados los espectros y realizado el cálculo de las cajas, se escalan al tamaño de la imagen original y se segmenta esa área en la imagen original.

Los espectros recortados son el primer producto científico, i.e. espectro bidimensional con los metadatos (usualmente llamado "dato crudo"). Este nivel de producto se irá incorporando al repositorio público de datos astronómicos del SEDICI a través del proyecto ReTrOH y quedando a disposición de toda la comunidad.

3. Software

Finalmente, se desarrolló un software que permite cargar una imagen escaneada y encontrar de forma automática la región que incluya cada uno de los conjuntos espectrales. Luego, el usuario puede verificar la selección (visualmente) y, una vez aceptada la región, permite introducir los metadatos correspondientes. Por último, se puede almacenar de forma individualizada cada uno de los espectros detectados, con sus metadatos asociados en un archivo FITS.

Para realizar el software se utilizaron buenas prácticas de programación y programación orientada a objetos. Las tecnologías utilizadas incluyen el entorno de trabajo de Svelte[†] para la interfaz de usuario, (*front end*), el cual está escrito en javascrip. Para el servidor (*back end*) se utilizó Flask (Grinberg, 2018) el cual es un marco de aplicaciones web (*microframework*), escrito en python. Se decidió trabajar con Flask porque puede ser desarrollado para cumplir la función de brindar una API (*Application Programming Interface*) robusta al *back end*. Mientras que para el *front end* se eligió trabajar con Svelte para generar interfaces más dinámicas para el usuario. El software tiene las siguientes funcionalidades:

- Al cargar una imagen se detectan de forma automática los espectros presentes.
- Se puede agrandar la imagen, y el usuario puede modificar el área detectada si fuera necesario.
- Permite agregar y/o quitar áreas de detección.
- Muestra la información detallada de la imagen cargada.
- Muestra un formulario por cada espectro detectado, en el cual se cargan los metadatos.
- Permite guardar en formato FITS los espectros detectados con sus respectivos metadatos.

Agradecimientos: Este proyecto ha recibido financiación de la Asociación Argentina de Astronomía a través de la Beca de Servicios Tipo A asignada a Nehuén Pereyra para realizar el desarrollo del software, y del Proyecto 11/G167 de la UNLP.

Referencias

Astropy Collaboration, et al., 2013, A&A, 558, A33
Astropy Collaboration, et al., 2018, AJ, 156, 123
Astropy Collaboration, et al., 2022, apj, 935, 167
Grinberg M., 2018, *Flask web development: developing web applications with python*, .ºReilly Media, Inc."
Jocher G., 2022, Yolov5
Kluyver T., et al., 2016, F. Loizides, B. Schmidt (Eds.), *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, 87 – 90, IOS Press
Meilán N., et al., 2020, BAAA, 61B, 251
Meilán N.S., et al., 2022, Epistemología e Historia de la Astronomía. Volumen I, I, 211
Meilán N., 2018, Recuperación del patrimonio observacional histórico. elaboración de un método óptimo de digitalización y extracción de datos
Redmon J., et al., 2016, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788

[†]https://svelte.dev/