ASOCIACIÓN ARGENTINA DE ASTRONOMÍA
BECA DE SERVICIO TIPO A

Desarrollo de software para la digitalización de placas espectrográficas

Candidato:
Nehuen Pereyra

Directora:
Dra. Yael Aidelman
Codirector:
Dr. Franco Ronchetti

Contacto: nehuenpereyra@gmail.com

Índice

1. Introducción

El trabajo se desarrolló en el marco de la beca de servicio tipo A, la cual consistió en mejorar la eficiencia del proceso de digitalización de espectros estelares almacenados en placas de vidrio. Para ello, se desarrolló un software que permite asistir a un usuario en esta tarea y de este modo minimizar la posibilidad de errores humanos como también del software ya que el usuario tiene la posibilidad de hacer modificaciones sobre las predicciones realizadas. El software realiza la detección de forma automática (con posibilidad de asistencia manual) e individualización de cada espectro registrado en las placas y el agregado de los metadatos correspondientes.

2. Trabajo realizado

2.1. 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.

Bajo este marco, la Facultad de Ciencias Astronómicas y Geofísicas de la UNLP, a través del proyecto ReTrOH (Recuperación del Trabajo Observacional Histórico)[1], se encuentra realizando un proceso de digitalización de una gran colección de placas espectroscópicas en formato de vidrio[2]. De este modo es posible el almacenamiento perpetuo de información adquirida durante casi un siglo y su posterior procesamiento.

Por otro lado, las Redes Neuronales son los modelos de aprendizaje automático con mejor desempeño en la actualidad en 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*)[3].

En este contexto, el procesamiento automático de las placas espectrográficas, detectando los espectros de ciencia individuales que en estas hubiera es un gran aporte para la comunidad astronómica.

2.2. Análisis de las placas espectrográficas

En la primera etapa se analizaron dos colecciones digitalizadas de placas espectroscópicas. La primera corresponde a una porción de la colección de Virpi S. Niemelä[4], esta abarca un total de 111 placas espectroscópicas, las cuales están disponibles en el repositorio institucional de la UNLP, SEDICI (Servicio de Difusión de la Creación Intelectual)[5]. Esta colección de placas espectroscópicas, se encuentra en formato FITS (*Flexible Image Transport System*). Éste es un formato estándar utilizado en la comunidad astronómica, ya que permite almacenar la imagen de la observación realizada y una cabecera (*header*) con los metadatos de la imagen. La segunda colección, fue digitalizada en el ICATE (Instituto de Ciencias Astronómicas, de la Tierra y el Espacio, UNSJ-CONICET)[6] y consta de 154 imágenes formato TIF (*Tagged Image File Format*).

Para trabajar con el formato FITS se utilizó el lenguaje python y en específico la librería Astropy[7]. 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[8] como interfaz para visualizar los datos.

En la figura 1 se puede observar un espectro digitalizado, el cual posee el espectro de ciencia en el centro y las lámparas de comparación a sus lados.

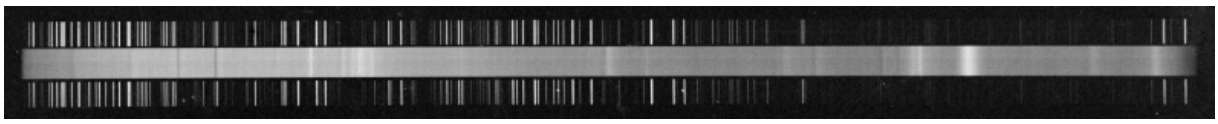

Figura 1: Ejemplo de un espectro digitalizado.

[1] https://retroh.fcaglp.unlp.edu.ar
[2] http://sedici.unlp.edu.ar/handle/10915/74678
[3] Francois Chollet. (22 Diciembre 2017). Deep Learning With Python. Manning Publications
[4] http://sedici.unlp.edu.ar/handle/10915/74499/discover
[5] http://sedici.unlp.edu.ar
[6] https://icate.conicet.gov.ar
[7] https://www.astropy.org
[8] https://jupyter.org

Los metadatos de esa placa espectrográfica se puede ver en la figura 2.

```
ORIGIN  = 'NOAO-IRAF FITS Image Kernel July 2003' / FITS file originato
DATE    = '2019-06-04T19:14:28' / Date FITS file was generated
IRAF-TLM= '2019-10-17T18:35:09' / Time of last modification
DATAMIN =                    0. / Minimum data value
DATAMAX =                65535. / Maximum data value
OBJECT  = 'HD 163758'
OBSERVAT= 'ctio    '
DATE-OBS= '1976-04-10'
TIME-OBS=                 7.25
UT      =                 7.25
ST      =             15.77215
HA      =             21.83715
RA      =              17.935
DEC     =             -36.0183
EPOCH   =               1976.3
RA2000  =             17.99121
DEC2000 =              -36.021
RA1950  =              17.935
DEC1950 =             -36.0183
EXPTIME =                960.
TELESCOP= '36 Inch '
DETECTOR= 'Photographic Plate'
GAIN    = '       '
RDNOISE = '       '
IMAGETYP= 'object  '
OBSERVER= 'Virpi Niemela'
INSTRUME= 'Cassegrain Spectrograph'
COMMENT = 'emulsion IIa0'
COMMENT1= 'slit 6/.12'
COMMENT2= 'viento, cirrus'
COMMENT3= '       '
DIGITALI= 'Rosario Alessandroni'
SCANNER = 'Nikon 9000ED'
SOFTWARE= 'VueScan 9 x64 (9.5.81)'
PLATE-N = 'A4430   '
SPEC-AMO=                    3
SPEC-ID = 'A(right) B(middle) C(left)'
SPEC-POS= 'A       '
MAIN-ID = 'HD 163758'
SPTYPE  = 'O6.5Iafp'
```

Figura 2: Ejemplo de los metadatos cargados en un archivo FITS.

La información almacenada en los metadatos le permite a los astrónomos identificar el espectro, ya que la colección de Virpi S. Niemelä en una sola imagen puede haber hasta 4 espectros en la misma imagen. Esto se puede ver en la figura 3.

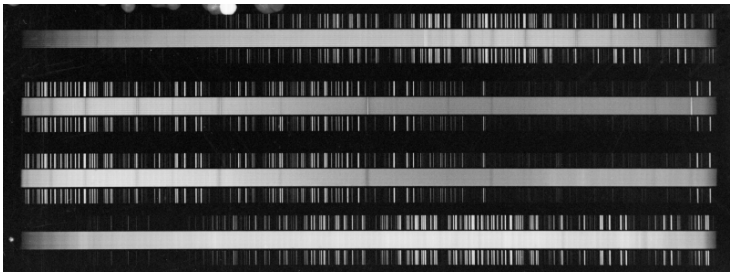

Figura 3: Ejemplo de una placa espectrográfica digitalizada.

El siguiente paso tratará de automatizar el recorte e individualización de cada espectro registrado en las placas y agregar los metadatos correspondientes. De este modo, se obtendría el primer producto científico, i.e. espectro bidimensional con los metadatos (usualmente llamado "dato crudo"). Este nivel de producto sería incorporado al repositorio público de datos astronómicos del SEDICI a través del proyecto ReTrOH y así quedará a disposición de toda la comunidad.

2.3. Recorte de los espectros de forma automática

La segunda etapa automatizó el recorte e individualización de cada espectro registrado en las placas. Para ello se utilizó un entorno de trabajo escrito en python, llamado YOLO (*You only look once*)[9] el cual permite realizar la detección de objetos, utilizando técnicas de aprendizaje profundo para entrenar un modelo y luego poder detectar los objetos de interés en una imagen o video.

Para entrenar el modelo de aprendizaje profundo se utilizaron las imágenes de los archivos FITS en formato png y se les redujo tanto la profundidad de color (cantidad de bits de información necesarios para representar el color de un píxel en una imagen digital) como las dimensiones (ancho y alto de la imagen digital) para acelerar el proceso de entrenamiento. El objetivo, es que luego de detectado el espectro, el recorte se realice sobre la imagen original (se requerirá una transformación del área detectada).

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

2.3.1. 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[10] el cual es de código abierto y permite exportar a múltiples formatos los datos etiquetados (incluido el formato que utiliza YOLOv5[11]). Con este software,

[9] https://pjreddie.com/media/files/papers/yolo_1.pdf
[10] https://labelstud.io
[11] https://github.com/ultralytics/yolov5

se etiquetaron las 111 placas espectroscópicas correspondientes a una porción de la colección de Virpi S. Niemelä y las 154 de la colección del ICATE. En la figura 4 se observa una placa espectroscópica etiquetada con el software mencionado, donde los recuadros verdes corresponden al etiquetado.

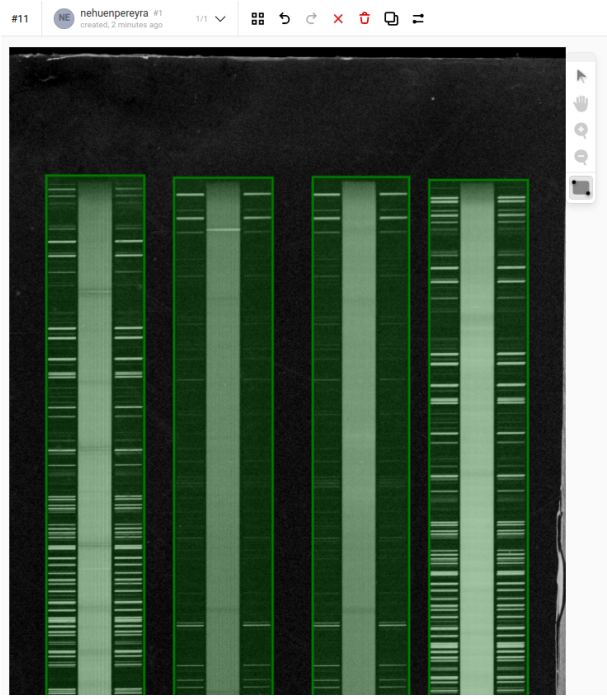

Figura 4: Etiquetado de una placa espectroscópica digitalizada.

2.3.2. Entrenamiento del modelo

Para llevar a cabo el entrenamiento del modelo, se separó el conjunto de datos en dos: el conjunto de entrenamiento el cual se utiliza para entrenar al modelo y un conjunto de evaluación el cual se utiliza para verificar cómo funciona el modelo con datos que nunca vio.

Para que el modelo tenga más datos para entrenar y pueda "aprender" a identificar los espectros de mejor manera, se utilizó la técnica de *data augmentation* la cual consiste en darle al modelo imágenes ligeramente modificadas (de forma aleatoria) respecto a las originales. Por lo tanto, nunca aprende en base a las imágenes originales, sino en base a las modificadas, permitiendo agrandar el conjunto de entrenamiento. YOLOv5 tiene esta técnica integrada en el propio entorno de trabajo y fue utilizado para configurar las modificaciones.

Se definió para el entrenamiento, un total de 250 épocas con un tamaño de lote de 64. La cantidad de épocas permite indicar la cantidad de iteraciones de entrenamiento, en relación al tamaño del conjunto de datos. Como entrada al modelo a entrenar no se puede enviar todo el conjunto de datos, se debe dividir en bloques llamados tamaño de lote, en general el tamaño de este es una potencia de 2.

Para realizar el entrenamiento, se realizaron las siguientes configuraciones:

- Se estableció un 90 % del conjunto de datos para el conjunto de entrenamiento y un 10 % para el conjunto de evaluación.
- 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 utilizo un tamaño de imagen de 512 pixeles.
- Se entrenó el modelo en un periodo de 250 épocas con lotes de 64.

Luego del entrenamiento, se analizaron las predicciones realizadas por el modelo y las métricas obtenidas. En la figura 5, se pueden observar las predicciones realizadas por el modelo.

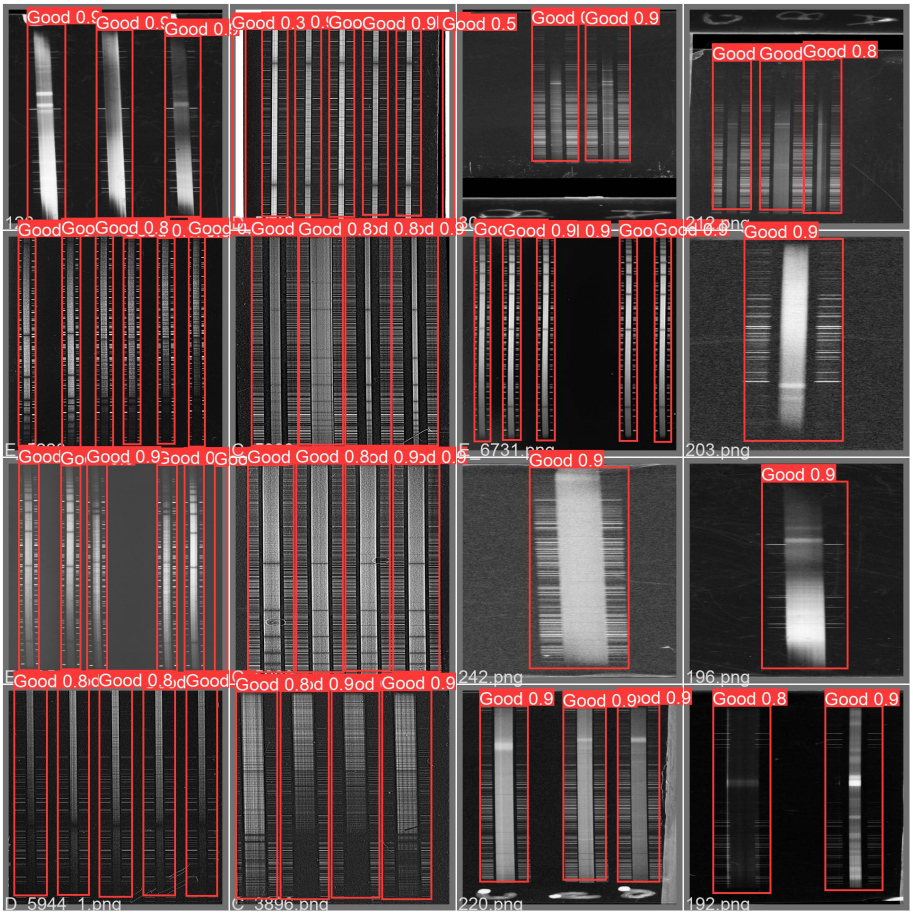

Figura 5: Predicciones del modelo entrenado.

Como se pueden observar las predicciones reflejan el resultado esperado. Para analizar correctamente lo que aprendió el modelo, es necesario observar las métricas de precisión (*precision*) y sensibilidad (*recall*). La precisión indica cuántos ítems reconocidos son realmente relevantes. Mientras que la sensibilidad 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. En la figura 6 se pueden observar las métricas de precisión y sensibilidad obtenidas luego del entrenamiento.

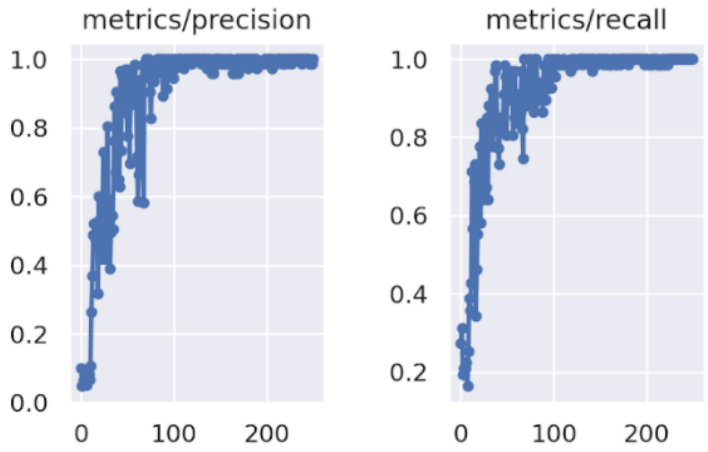

Figura 6: Métricas de precisión (*precision*) y sensibilidad (*recall*) del entrenamiento.

Se pueden observar que se tiene un 0.99 de precisión y de sensibilidad por lo tanto el modelo ha logrado generalizar y detectar los espectros en las imágenes.

2.3.3. Recorte automático

En esta última etapa, se realiza el recorte de forma automática de los espectros detectados en una imagen. Para ello, primero se extrae del archivo FITS la imagen original, luego se la transforma a un tamaño de 512 x 512 píxeles (ya que el modelo entrenado trabaja con esas dimensiones) y se la utiliza como entrada del modelo entrenado. Como salida se obtienen las cajas de las predicciones (bbox), cada bbox identifica a cada espectro detectado. Las bbox están conformadas por los cuatro vértices del área detectada y a la clase que pertenece el objeto detectado (en este caso solo existe una clase, que es la de espectro). Una vez detectados los espectros y realizado el cálculo de los bbox, se escalan al tamaño de la imagen original y se recorta esa área en la imagen original utilizando python.

Al realizar la individualización de los espectros de una placa espectrográfica, en general no se tienen los metadatos previamente cargados. Por lo tanto, se realizará un recorte semiautomatico en donde un software asistirá al astrónomo detectando los espectros y este confirmará si se detectó correctamente o podría hacer algunas modificaciones del área detectada, para luego cargar los metadatos correspondientes a ese espectro.

2.4. Software

En la última etapa de la beca, 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 de forma ocular 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 fueron, el entorno de trabajo de Svelte[12] para el *frontend*, el cual está escrito en javascript y como *backend* se utilizó Flask[13] el cual es un *microframework* que está escrito en python. Se escogió trabajar con Flask, porque al ser un *microframework* puede ser desarrollado para cumplir la función de brindar una API (*Application Programming Interface*) robusta al *frontend*. Mientras que para el *frontend* se eligió trabajar con Svelte para generar interfaces más dinámicas para el usuario.

En la figura 7 se puede ver la interfaz del software con los espectros detectados en la imagen.

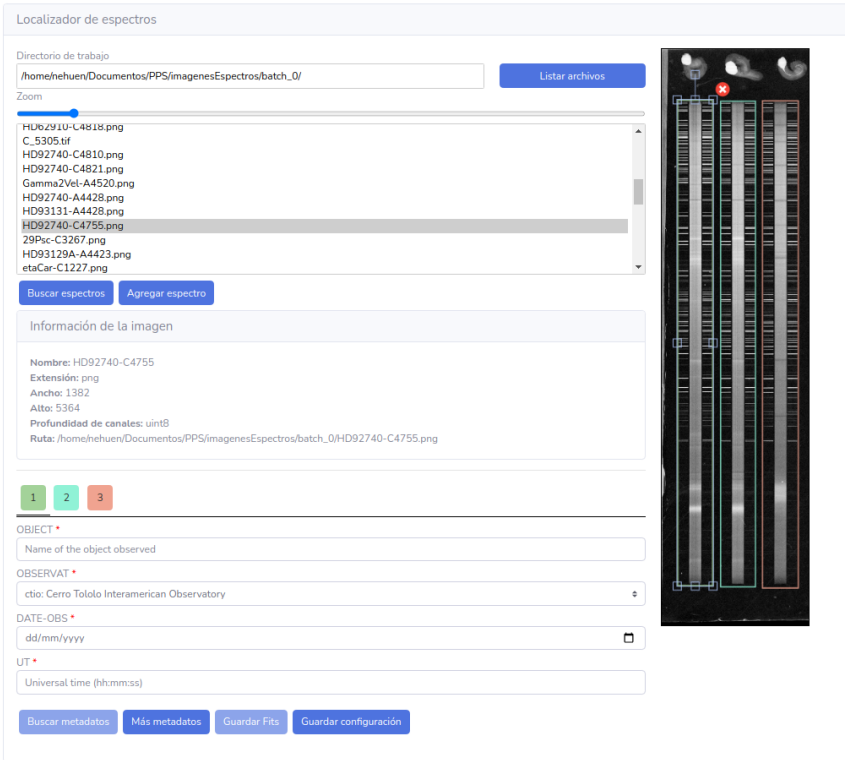

Figura 7: Interfaz gráfica del software de detección de espectros.

[12] https://svelte.dev
[13] https://flask.palletsprojects.com/en/2.0.x/

Como se puede observar, el software tiene las siguientes funcionalidades:

- Al cargar una imagen, se detectan de forma automática los espectros que aparecen.
- Se puede hacer *zoom* a la imagen, ya que el usuario puede modificar el área detectada, para mejorar la precisión de la detección si es necesario.
- Permite agregar y quitar áreas de detección.
- Se muestra la información detallada de la imagen cargada.
- Se 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.

3. Logros obtenidos

Se logró cumplir con el objetivo de la beca, el cual consistía en desarrollar un software que permita agilizar el proceso de digitalización de las placas espectrográficas. El modelo entrenado de inteligencia artificial realiza buenas predicciones a la hora de la detección individualizada de los espectros estelares. De este modo, el software facilita tanto el recorte como la carga de los metadatos asociados a un espectro estelar individual.

Como conclusión, se desarrolló un software que es de utilidad para la comunidad astronómica, el cual quedará con una licencia GNU GPL (Licencia Pública General de GNU) para que otros profesionales puedan usar, estudiar, compartir (copiar) y modificar el software.

4. Perspectivas futuras

Las imágenes FITS con su cabezal (*header*) correspondiente, obtenidas al utilizar el software estarán listas para ser subidas a la base de datos del SEDICI, repositorio oficial de la UNLP que se está utilizando para los productos recuperados por ReTrOH.

La etapa siguiente a la digitalización sería automatizar los procesos de extracción y calibración en longitud de onda de los espectros estelares. Estos espectros unidimensionales, serán subidos al NOVA (Nuevo Observatorio Virtual Argentino)[14].

[14] https://nova.conicet.gov.ar/