
Miniproyecto 1:

Introducción al procesamiento de imágenes mediante un problema de detección

1. Objetivos

- Familiarizarse con los componentes principales de una base de datos de visión por computador.
- Conocer y aprender a utilizar las librerías básicas de Python para el análisis y procesamiento de imágenes.
- Procesar las imágenes de un directorio de manera global.
- Aplicar las herramientas teóricas y la metodología de trabajo adecuada para abordar un problema de detección, una de las tareas principales en la visión por computador.

2. Parámetros de entrega

No se recibirá ningún archivo por algún medio diferente a Brightspace.

PRE-ENTREGAS: Son de carácter individual y tendrán un peso asignado dentro de la calificación de su respectiva entrega. En cada una de las pre-entregas deberán adjuntar un único archivo *.pdf* nombrado de la siguiente manera: *preentrega_CódigoEstudiante.pdf*. El plazo de estas pre-entregas será hasta las 8:59 a.m. de las fechas previstas:

- Pre-entrega 1: Lunes 07 de febrero de 2022
- Pre-entrega 2: Lunes 14 de febrero de 2022
- Pre-entrega 3: Lunes 28 de febrero de 2022

ENTREGAS:

- Adjuntar un único archivo con los códigos de la siguiente forma:
main_CódigoEstudiante1_CódigoEstudiante2.py.
 - **NO** se recibirán archivos con otra extensión. Su entrega conllevará una penalización sobre la calificación final de la entrega.
 - **La entrega de un archivo .py es obligatoria.**
 - El código enviado debe tener una longitud máxima de 500 líneas. Se generará una penalización sobre la calificación final de cada entrega por cada 10 líneas adicionales. Esto con el objetivo de promover el uso de estructuras iterativas y operaciones vectoriales, que aumenten la eficiencia de sus códigos (aspecto fundamental en tareas de programación).
- Adjuntar una copia de **TODO** el código .py del ítem anterior en formato .txt. Este debe llamarse de la siguiente manera: *CódigoEstudiante1_CódigoEstudiante2.txt*. Esto con el fin de poder evaluar en Brightspace automáticamente cualquier intento de copia, plagio, o similitud entre los algoritmos. **Cualquier intento de copia/plagio será tratado de acuerdo al reglamento de la Universidad. Aquel grupo que no incluya este ítem en cualquiera de sus entregas tendrá una nota de 0 en la entrega que corresponda.**
- Adjuntar un único archivo con el informe en PDF. El informe debe presentarse con el formato CVPR (para más información del informe, ver sección 4). Este debe ser nombrado de la siguiente forma: *CódigoEstudiante1_CódigoEstudiante2.pdf*.
- En el caso de que no tenga compañero, debe utilizar un cero como el código de su compañero, i.e. *main_CódigoEstudiante_0.py* para el archivo de código, *CódigoEstudiante_0.pdf* para el informe y *CódigoEstudiante_0.txt* para el archivo de texto.
- **NO** se permiten archivos comprimidos tales como *zip*, *rar*, *7z*, *tar*, *gz*, *xz*, *iso*, *cbz*, etc. Aquel grupo que envíe su entrega como un archivo comprimido tendrá una penalización sobre la calificación de la entrega.
- Este proyecto tiene una duración de cuatro semanas y está compuesto por tres entregas. Cada entrega del presente miniproyecto recibirá una calificación por separado. Estas se entregarán en las siguientes fechas:
 - Entrega 1 (3 %): Viernes 11 de febrero de 2022
 - Entrega 2 (3 %): Viernes 18 de febrero de 2022
 - Entrega 3 (4 %): Viernes 04 de marzo de 2022

El plazo de entrega será hasta las 9:59 p.m. de las fechas establecidas. Sin embargo, el vínculo para el envío de los documentos estará disponible hasta las 10:59 p.m. De 10:00 p. m a 10:29 p.m la entrega se calificará sobre 4.0. Y de 10:30 p.m a 10:59 p.m la entrega se calificará sobre 3.0 (tenga en cuenta que la hora queda registrada en el envío de Brightspace).

- Se tendrá flexibilidad únicamente ante las circunstancias extraordinarias y estipuladas en el reglamento de la Universidad. Ante esta situación, les pediremos se comuniquen con una anticipación moderada (no a última hora antes de las entregas) con los dos asistentes docentes por correo explicando su situación y así poder establecer un nuevo plazo para su entrega por Brightspace.
- Cada estudiante tiene varios intentos en Brightspace para cada entrega, pueden hacer entregas parciales y **se tendrá en cuenta únicamente el último intento** para la calificación.
- Teniendo en cuenta que se recibirán archivos después de la hora estipulada con una penalización, **no se recibirán documentos o archivos después de la hora máxima de entrega (10:59 p.m) ni por Brightspace, ni por ningún otro medio.**

3. Reglas Generales

- La asistencia a la sección de laboratorio es **obligatoria**. De acuerdo con el Reglamento General de Estudiantes de Pregrado de la Universidad de los Andes, la inasistencia a más del 20 % de las clases de laboratorio resultará en la reprobación de la totalidad de la materia (laboratorio y magistral).
- Los informes deben realizarse **únicamente** con la pareja. Esto quiere decir que, aunque es válido discutir los problemas generales con sus compañeros, la solución y el código deben ser de su completa autoría. **Está prohibido** copiar literalmente el algoritmo y/o procedimiento desarrollado por otro grupo.
 - Si llega a obtener un código de Internet, asociado al problema a resolver, este debe estar **debidamente referenciado** y usted debe entenderlo por completo. Sin embargo, todos los ejercicios que requieren una implementación propia, deben ser de su completa autoría. En estos casos, no podrán tomar implementaciones (o partes) de un autor externo así estén adecuadamente referenciadas.

4. Parámetros de Calificación

Resultados:

1. Toda figura debe estar numerada y debe llevar su título y descripción en el informe.
2. Todos los códigos deben mantener orden y coherencia en la ejecución de comandos. Cada vez que se muestre una figura, el programa debe esperar para que se presione una tecla, para así continuar con la siguiente y así sucesivamente (para esto utilice en Python `input("Press Enter to continue...")`). Si quiere contrastar dos o más imágenes utilice un subplot.
3. Independientemente del entorno de desarrollo integrado (IDLE) que utilicen para implementar sus códigos con extensión .py, solo utilicen funciones y librerías propias de python. Por lo que, se debería poder correr su código desde cualquier compilador o terminal.

4. Dado que esta es una materia de programación, es igual de importante tanto la sintaxis como la ejecución de sus códigos. Por este motivo, la nota correspondiente al código será distribuida entre la correcta ejecución del código y el desarrollo del mismo.
5. El código debe estar adecuadamente documentado (comentado).
6. Para no tener inconvenientes a la hora de realizar la entrega, asuma que al mismo nivel de ejecución del código se encuentra una carpeta con **TODOS** los archivos necesarios para el laboratorio. Esta carpeta **SIEMPRE** será proveída por lo que no deben cambiar ni el nombre de la carpeta ni de los archivos dentro de ella.
7. No utilice rutas absolutas para leer o guardar archivos. Este es el error más común en la ejecución de los códigos.

Para generar rutas utilice `os.path.join` en Python, ya que puede que corramos los laboratorios usando Linux o Windows y los separadores de archivos cambian dependiendo del sistema operativo.

Ejemplo: Dentro del código principal, el estudiante quiere leer la imagen `im.png` que está dentro de una carpeta de imágenes en la misma ruta que el `main`.

7.1. Forma incorrecta:

```
skimage.io.imread('C:/Estudiante/docs/ElProyecto/ims/im.png')
```

7.2. Forma correcta:

```
skimage.io.imread(os.path.join('ims', 'im.png'))
```

Informe:

Todos los laboratorios deben realizarse en formato *L^AT_EX* o Word, pueden obtener una plantilla del formato en el siguiente *vínculo*. Cabe resaltar que los informes no deben contener ninguna sección de artículo científico, esto significa que no deben incluir ninguna división como introducción, resultados, *abstract* o conclusiones. Por consiguiente, las secciones o ítems incluídos deberán estar destinados a responder únicamente las preguntas del informe de forma organizada, para incluir las imágenes requeridas de sus resultados y documentarlas debidamente. Por último, cada informe tiene una **longitud máxima de 4 páginas**, excluyendo referencias y anexos. Se pueden incluir imágenes y/o tablas en la sección de Anexos pero los resultados principales (solicitados) deben ser parte del cuerpo del informe.

Bonos:

Cada pareja podrá obtener puntos adicionales por cumplir la siguiente característica:

1. Desarrollar el informe en *L^AT_EX*. Aquellas personas que lo desarrollen en *L^AT_EX*, deben escribir al final del informe **”Realizado en *L^AT_EX*”**. De lo contrario no se contará el bono. Para poder escribir el logo utilice el comando “`\LaTeX`” en su editor de Latex.

Estos puntos se asignarán de acuerdo al criterio de los profesores y los grupos que intenten reproducir la frase en un informe realizado en Word obtendrán cero (0) en la nota de dicha entrega.

5. Información General

5.1. Tipos de Imágenes

Existen cuatro tipos de imágenes básicas, cuyas descripciones se encuentran a continuación. Ustedes podrán explorar estos tipos en este proyecto y a lo largo del semestre.

Cuadro 1: Tipos de imágenes

Tipo de imagen	Definición
Binaria	Una matriz lógica, sólo contiene unos y ceros, interpretados como blancos y negros respectivamente.
Indexada	También conocida como una imagen en pseudocolor. Matriz de clase logical, uint8, uint16, single o double. Los pixeles son índices dentro de un mapa de colores. El mapa de colores es una matriz de tipo double de dimensiones Mx3.
Escala de grises	Se conoce también como imagen de intensidades. Matriz de tipo uint8, uint16, int16, single o double, donde los pixeles especifican los valores específicos de la intensidad.
Color	También conocida como imagen RGB (o en otro espacio de color). Matriz MxNx3 para la que en cada pixel se interpretan las intensidades de los canales como un color.

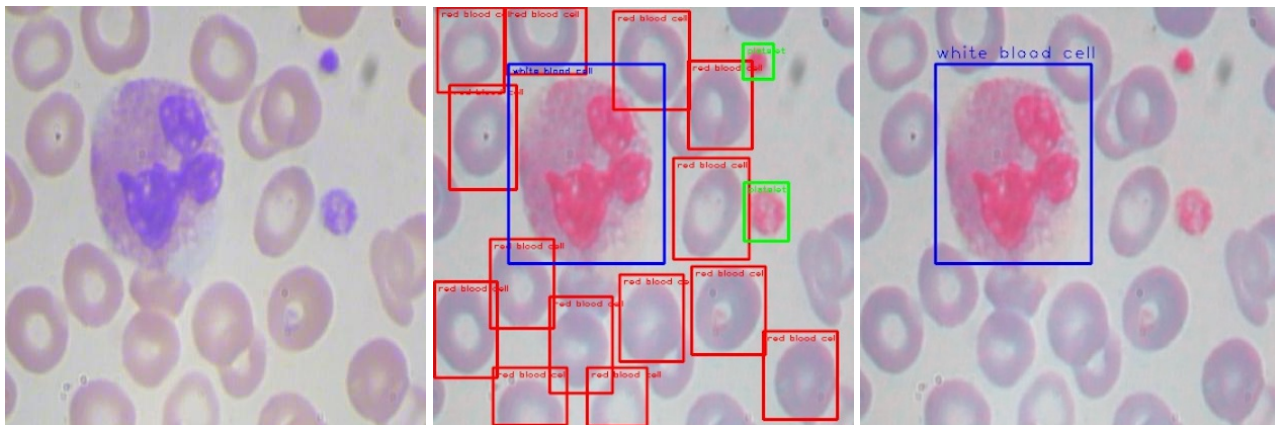
5.2. Paquetes especiales a importar

1. La librería `os`
2. La librería `shutil`
3. La librería `glob`
4. La librería `cv2`
5. La librería `scikit-image`

6. Definición del problema

La detección automática de objetos ha sido un problema longevo en el campo de la visión por computador. Esta tarea será el eje central de este primer miniproyecto donde resolveremos de forma parcial el problema planteado por el “BCCD Dataset”. En nuestro caso utilizaremos una versión modificada del dataset original, el cuál se encuentra disponible en Roboflow. **Para la totalidad de este miniproyecto podrá encontrar la base de datos modificada que utilizaremos y todos los recursos necesarios dentro de la carpeta data_mpl.**

La tarea propuesta en la base de datos es la de detección de células sanguíneas. Sin embargo, como ya se les mencionó, a lo largo de este miniproyecto nos centraremos únicamente en una sección de esta problemática. El problema puntual que será resuelto es el de detección de glóbulos blancos teniendo en cuenta las anotaciones proveídas por el dataset. En la Figura 1 se muestra un ejemplo para la visualización de las anotaciones originales que presenta el dataset para una imagen así como una visualización para la anotación que nos es de interés para nuestro problema acotado.



(a) Imagen de entrenamiento

(b) Anotaciones completas

(c) Anotación de interés

Figura 1: Ejemplo de una imagen de entrenamiento del dataset, una visualización de las anotaciones entregadas y una visualización únicamente de las anotaciones de interés.

A lo largo de este miniproyecto realizarán una primera aproximación a la resolución de un problema biomédico con herramientas de visión por computador. Lo harán de forma progresiva, en primer lugar se familiarizarán con la base de datos al explorar las imágenes y las anotaciones. Luego, implementarán una estrategia de evaluación para predicciones del problema en cuestión. Finalmente, harán uso de una metodología experimental para determinar la mejor estrategia para resolver este problema al usar las técnicas que tienen a su disposición hasta el momento. Esta experimentación será principalmente llevada a cabo alrededor de factores como el pre-procesamiento, la técnica de umbralización a utilizar para segmentar y técnicas de post-procesamiento.

Al momento de abordar el problema de detección, esta se puede realizar principalmente por dos métodos diferentes. La aproximación de ventana deslizante y la aproximación por segmentación. En este miniproyecto abordaremos la tarea mediante la segunda aproximación, por lo cuál como parte del proceso deberán segmentar los glóbulos blancos y en base a estas máscaras de segmentación definirán las predicciones de detección que serán evaluadas.

Para realizar la segmentación haremos uso de herramientas básicas de la visión por computador como lo son las transformaciones de intensidad y la umbralización, además de operaciones entre imágenes binarias. Una vez obtengan la segmentación deseada, harán dos tipos de procesamientos (mediante transformaciones geométricas y mediante técnicas relacionadas con componentes conexos) para así poder eliminar las secciones indeseadas que se incluyeron en las segmentaciones obtenidas. Finalmente, harán las predicciones de detección a partir de las segmentaciones, validarán los resultados de los diferentes experimentos y obtendrán una evaluación del método final. Para esto, deberán aprender a utilizar adecuadamente los diferentes “splits” de la base de datos; los cuáles son: imágenes de entrenamiento (*train*), validación (*valid*), y de prueba (*test*).

7. Procedimiento Entrega 1

En esta entrega tendrán un primer acercamiento a los datos que utilizarán a lo largo de todo el miniproyecto. Para esto importarán, explorarán y visualizarán tanto las imágenes como las anotaciones. Y por último, aprenderán sobre una técnica importante para abordar problemas en visión por computador, el *data augmentation*.

7.1. Pre-entrega

Esta está basada en lo que desarrollarán en la sección 7.4. Las técnicas de *data augmentation* generalmente se usan cuando los datos que se tienen son pocos o no son representativos de la muestra poblacional que se desea analizar. Dos tipos importantes de aumento de datos son los de forma y los de color. Por medio de los de forma se modifica la geometría de la imagen para así obtener objetos en diferentes orientaciones y posiciones. Por otro lado, las transformaciones de color modifican la saturación, el brillo o agregan ruido a la imagen para poder incluir artificialmente imágenes con otras condiciones lumínicas o con diferente calidad. Como pre-entrega deberán buscar sobre los diferentes métodos de *data augmentation* (en especial acerca de las transformaciones geométricas y de intensidad) y justificar cuál o cuáles métodos serían útiles según la aproximación que se va a tomar en este problema.

7.2. Importación

Inicialmente harán la importación de los diferentes archivos y datos que usarán a lo largo de las entregas. Esto es necesario para acceder a ellos, procesarlos y manejarlos de forma correcta para así poder realizar el procedimiento adecuadamente.

7.2.1. Importación de las imágenes

Deben realizarlo para la totalidad de las imágenes de forma automática, es decir no deben importar las imágenes una por una usando su respectivo nombre. Para esto pueden usar la librería glob y almacenar las imágenes de cada split en una lista o arreglo diferente.

Una vez importadas las imágenes:

- Construyan un subplot como el de la Figura 2 donde se muestren 8 imágenes diferentes de la base de datos, incluir esta figura en su informe.
- Utilizando como soporte la figura anterior, en su informe respondan:
 - ¿Qué elementos contienen las imágenes?
 - ¿Cómo pueden diferenciar los glóbulos blancos de los rojos?
 - ¿Qué dificultades encuentran para diferenciar los glóbulos blancos de las plaquetas y en qué se diferencian?

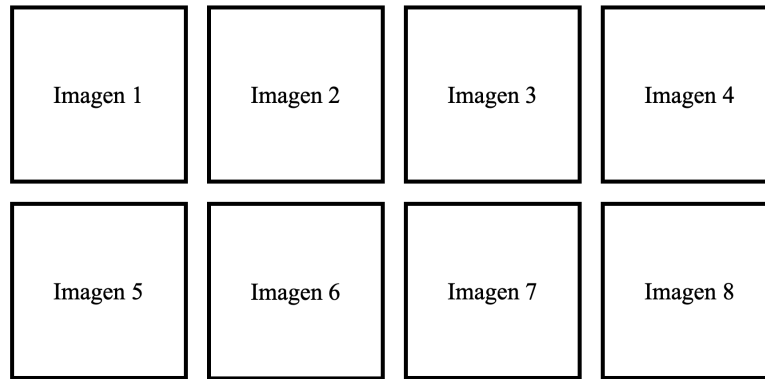


Figura 2: Ejemplo de subplot para la visualización de las imágenes.

7.2.2. Importación de anotaciones

Después de haber explorado las imágenes, deben abrir los archivos con extensión *.json* que contienen las anotaciones de detección. En consola deberán imprimir:

- La cantidad de anotaciones de detección que hay de cada clase en cada split y en total.
- El promedio de anotaciones por imagen para cada una de las clases en cada split y en total.

En su informe:

- Incluyan una tabla que condense la información mencionada.
- Respondan: ¿Cómo están organizados los archivos *.json*? ¿Cómo son las anotaciones? ¿Qué información incluye cada anotación?

7.2.3. Estadísticas

Incluyan en su informe:

- La información más relevante sobre la base de datos. Para esto incluyan la distribución que se tiene en las 3 diferentes carpetas (splits), apóyense de las estadísticas calculadas de las anotaciones y de calcular estadísticas adicionales también imprímanlas en consola.
- Respondan: ¿Consideran que la base de datos está correctamente distribuida? ¿Por qué?

7.3. Visualización y exploración

Ahora que tienen manejo los diferentes elementos que se usarán para el problema, es necesario poder visualizar estos datos y explorarlos en conjunto. Esto con el objetivo de comprender de mejor manera lo que se está buscando y qué características tienen las anotaciones al observarse sobre las imágenes. Además de esto, es necesario hacer una primera exploración de las imágenes para empezar a pensar en como aproximar el problema.

7.3.1. Visualización de Anotaciones

Observarán las anotaciones para visualizar qué partes de los objetos se consideran en la detección. Para esto deben construir un subplot 4x4 como el de la Figura 3 donde la primera columna corresponde a las imágenes originales y las otras 3 columnas corresponden a las anotaciones de cada una de las 3 clases por separado. Incluyan esta figura en su informe y analicen su contenido.

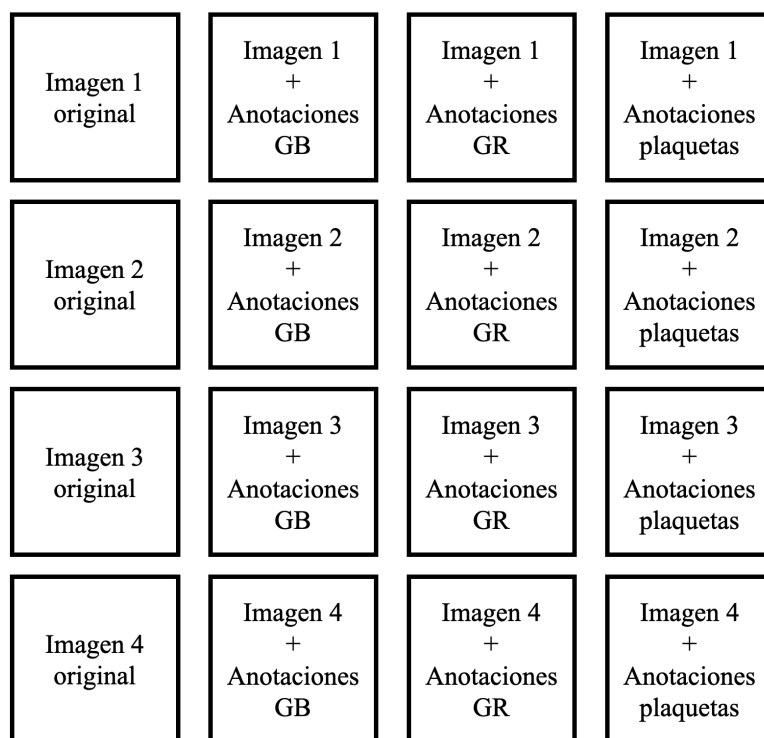


Figura 3: Ejemplo de subplot para la visualización de las anotaciones.

7.3.2. Exploración canales de color

Dado que realizarán la segmentación de los glóbulos blancos y que poseemos información valiosa en los tres canales de color de las imágenes, es fundamental realizar un análisis de estos. Para esto, deben realizar un subplot de 4x4 como el de la Figura 4 donde la primera columna corresponde a la imagen original y las otras 3 columnas corresponden a los diferentes canales que componen la imagen.

En su informe respondan:

- ¿Que representa cada uno de los canales?
- ¿Existe un canal que permita diferenciar de manera adecuada los glóbulos blancos?
- ¿Existe alguna forma en la que hacer una operación entre 2 canales logre obtener una diferenciación clara? ¿Cuáles propondrían?

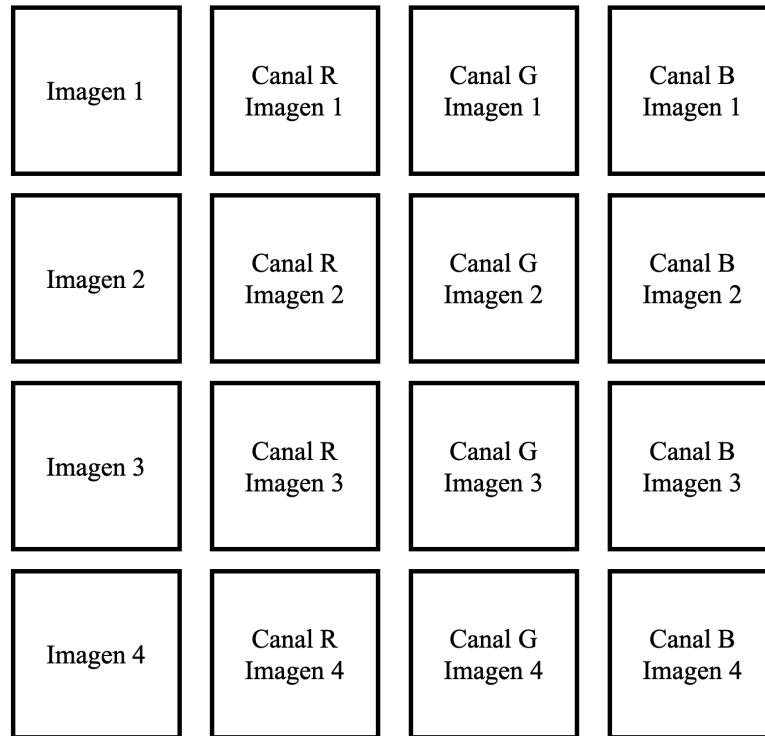


Figura 4: Ejemplo de subplot para la visualización de los canales de las imágenes.

7.4. Data augmentation

Finalmente, como última etapa de esta entrega harán lo que se conoce como *data augmentation*. Este es un proceso que suele usarse muy a menudo cuando las muestras son insuficientes para representar la población o se tienen muy pocas imágenes. Como se les mencionó en la pre-entrega, existen dos formas principales de aumento de datos: los de forma y los de color.

Discusión pre-entrega

A partir de lo que cada uno de ustedes investigó por su cuenta en la pre-entrega, discutan en el informe las diferencias entre sus entregas, cuáles transformaciones existen para hacer *data augmentation* y qué tipo de transformaciones podrían aplicar para este problema.

Implementación

A partir de lo discutido en la subsección anterior escojan 6 técnicas diferentes para realizar *data augmentation* (deben ser 3 geométricas y 3 de color). Deben aplicar estos tipos de transformaciones a una imagen, sin embargo no deben hacer todas estas transformaciones de forma manual ya que pueden hacer uso de librerías como scikit-image para generar las nuevas imágenes. Además, un recurso importante en la realización de diferentes tipos de procedimientos son los foros abiertos y tutoriales sobre el tema. Para este caso, además de usar la librería pueden apoyarse en foros sobre el tema para guiarse tanto en las funciones que pueden usar como en procedimientos que no tienen una función determinada sino que deben realizar operaciones de forma manual. Incluyan en su informe una figura de una única imagen original de la base de datos junto a las 6 transformaciones que escogieron.

Nota: Recuerden que siempre que usen códigos ajenos a ustedes o se apoyen en recursos externos DEBEN referenciar las respectivas fuentes adecuadamente.

8. Procedimiento Entrega 2

En esta entrega harán la preparación para evaluar el problema en la última entrega del miniproyecto, esto con el fin de una vez se tenga el método que se quiere validar poder hacerlo correctamente. Para esto, en primer lugar, implementarán una función para generar predicciones de detección a partir de una máscara de segmentación. En segundo lugar implementarán una función que evalúe las detecciones para determinar sus “true positives (TP)”, “false positives (FP)”, “true negatives (TN)” y false negatives (FN)”. En tercer lugar, implementarán una última función la cuál genere una curva de precisión y cobertura a partir de los TP, FP, TN y FN. Finalmente, podrán comprobar el correcto funcionamiento de sus funciones utilizando ejemplos “de juguete”, ya sean máscaras generadas por ustedes o predicciones de detección entregadas por nosotros.

8.1. Pre-entrega

Esta se basará en la función que desarrollarán en la sección 8.2. La entrega será un diagrama de flujo donde especifiquen como harán el paso de las mascaras de segmentación a las anotaciones de detección. Asuman que la entrada de la función será una imagen binaria con unos (1) donde hay objetos de interés y ceros (0) en el fondo (objetos que no son de interés), y la salida un diccionario que entre otros datos contiene un vector de 4 posiciones, donde las primeras dos corresponden a las coordenadas de la esquina superior izquierda de la caja (x,y) y las otras dos posiciones corresponden al ancho y alto de esta (w,h). De la misma forma, generalice a cómo implementaría una función a la cuál por parámetro le entra un diccionario cuyas llaves son los nombres de las imágenes de un *split* y sus valores las máscaras de segmentación generadas; esta deberá retornar una lista de los diccionarios que contienen la información de las predicciones de todo el *split*.

Hint: Para ver la estructura de como se deben retornar las predicciones de detección observe las predicciones de prueba que tienen disponibles en la carpeta que les fue entregada en el archivo “*dummy_predictions.json*”.

8.2. Función predicciones de detección

Como se mencionó previamente, la aproximación a detección que usarán será a partir de segmentación. Debido a esto es necesario que implementen una función que a partir de una máscara de segmentación genere las predicciones propias de la detección. Esta deberá tener la siguiente estructura:

Sintaxis:

```
Mask2Detection_Codigo1_Codigo2(mask, img_id)
```

Parámetros de entrada:

- `mask`: Mascara binaria de la cual se extraerán las predicciones de anotación.
- `img_id`: Id de la imagen.

Respuesta:

- `detection`: Lista de diccionarios con las predicciones. Esta debe tener el mismo formato ejemplificado en el archivo “*dummy_predictions.json*” en la carpeta que se les entregó.

Validación

Para comprobar el correcto funcionamiento de su función construya 3 imágenes binarias de 50x50 que en su interior tengan uno o más círculos por imagen (Cada círculo debe ser de diferente tamaño y estar en diferente posición). Usando las 3 imágenes binarias diseñadas, elaboren un subplot de 3x2 como el de la Figura 5 donde se pueda visualizar la máscara de segmentación original en la primera columna y la anotación de detección correspondiente en la otra columna.

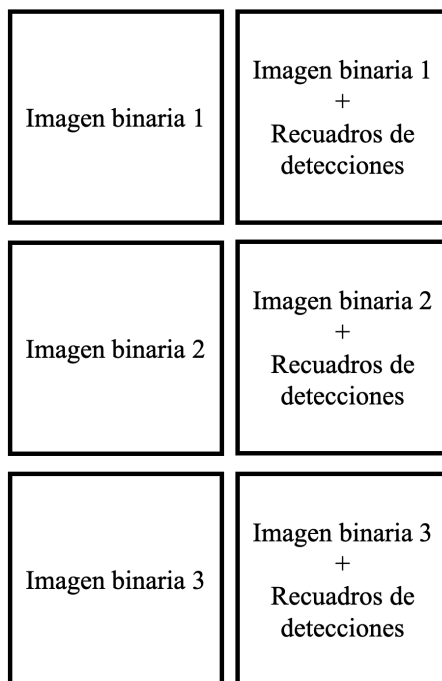


Figura 5: Ejemplo de subplot para visualizaciones de la función `Mask2Detection`.

8.3. Función evaluación predicciones

Ahora que ya tienen la capacidad de realizar las predicciones de detección deberán implementar una función para evaluarlas. Para esto, deben implementar una función que logre determinar la cantidad de TP, FP, TN y FN y tenga la siguiente estructura:

Sintaxis:

```
detections_Codigo1_Codigo2(conf_thresh, jaccard_thresh, annot_file, pred_file)
```

Parámetros de entrada:

- `conf_thresh`: Umbral de confianza a partir del cual tener en cuenta una detección.
- `jaccard_thresh`: Umbral del índice de Jaccard a partir del cual se debe considerar una detección como un verdadero positivo.
- `annot_file`: Ruta del archivo donde se encuentren las anotaciones.
- `pred_file`: Ruta del archivo donde se encuentren las predicciones.

Respuesta:

- TP: Cantidad de verdaderos positivos.
- FP: Cantidad de falsos positivos.
- TN: Cantidad de verdaderos negativos.
- FN: Cantidad de falsos negativos.

Validación

Para validar el funcionamiento de esta función deben utilizar los datos disponibles en el archivo *“dummy_predictions.json”* dentro de la carpeta *data_mp1* que les fue entregada. Con estos, y usando un umbral de confianza de 0.5 y un umbral del índice de Jaccard de 0.7, deben calcular las métricas especificadas en la función y presentarlo en su informe como una matriz de confusión junto con las métricas de precisión, cobertura y f-medida asociadas. Adicionalmente responda en su informe:

- ¿Qué se espera respecto a los falsos positivos y los falsos negativos al variar el umbral de confianza? ¿Por qué se espera este comportamiento?
- ¿Existe un umbral de confianza óptimo para usar en esta aplicación? ¿Cuál y por qué?
- ¿Qué información proveen los verdaderos negativos? ¿Tienen alguna utilidad? ¿Por qué?

8.4. Función curva de precisión y cobertura

La última función que deben implementar es una que en base a los archivos de anotaciones y de predicciones, el umbral deseado para considerar una predicción como correcta y la función realizada en el numeral anterior haga el cálculo de la curva de precisión y cobertura. Esta gráfica debe ser guardada en un archivo de imagen con la ruta indicada. Esta función debe tener la siguiente estructura:

Sintaxis:

```
PRCurve_Codigo1_Codigo2(jaccard_thresh, annot_file, pred_file, save_route)
```

Parámetros de entrada:

- *jaccard_thresh*: Umbral del índice de Jaccard a partir del cual se debe considerar una detección como un verdadero positivo.
- *annot_file*: Ruta del archivo donde se encuentren las anotaciones.
- *pred_file*: Ruta del archivo donde se encuentren las anotaciones.
- *save_route*: Ruta donde se vaya a guardar la imagen.

Respuesta:

- *area_under_the_curve*: El área por debajo de la curva de precisión y cobertura.
- *data*: Lista con precisiones y coberturas de todos los puntos de la curva.

Validación

Haciendo uso de todas las funciones implementadas deberán evaluar esta última usando los datos que tienen disponibles en la carpeta `data_mp1` en el archivo `“dummy_predictions.json”`. Usando estos datos realicen 3 curvas de precisión y cobertura e incluyanlas en una misma Figura. Para esto utilicen 3 índices de Jaccard diferentes [0.5, 0.7, 0.9]. Adicionalmente respondan en su informe: ¿Con qué propósito se varía el umbral del índice de Jaccard? ¿Con qué propósito se varía el umbral de confianza? ¿Qué diferencias pueden observar en las 3 curvas? ¿A qué se deben?

9. Procedimiento Entrega 3

En esta ultima entrega del miniproyecto utilizarán el método de segmentación y realizarán una experimentación detallada para seleccionar su método de detección final. Para esto harán un pre-procesamiento a los canales de las imágenes basándose en lo explorado durante la primera entrega del miniproyecto y utilizando transformaciones de intensidades. Teniendo en cuenta lo anterior, y habiendo seleccionado el pre-rocesamiento más adecuado, realizarán la segmentación y un post-procesamiento de esta con el fin de eliminar la presencia de artefactos (componentes de la segmentación no deseados). Luego, aplicando diferentes formas de realizar la segmentación y diferentes tipos de post-procesamiento harán la etapa de experimentación. Por último, realizarán la evaluación de su mejor método.

9.1. Pre-entrega

La pre-entrega correspondiente a esta última entrega del miniproyecto será un trabajo previo a lo que deberán implementar en la sección 9.3.2. Como podrán intuir, la tarea que realizarán será con el fin de mantener únicamente los componentes conexos de gran tamaño. Para esto harán un diagrama de flujo para explicar como realizarán el procedimiento para mantener únicamente los componentes más grandes basándose en la información de componentes conexos. Tengan en cuenta que la entrada debe ser la máscara de segmentación con artefactos y la salida debe ser la máscara sin ningún artefacto. Si por algún motivo necesitan más de un proceso para entrenar su algoritmo pueden utilizar más de un diagrama de flujo.

9.2. Pre-procesamiento

En esta primera sección de esta entrega harán un paso importante para la solución de un problema en la visión por computador, el cual es el pre-procesamiento. Aquí harán transformaciones a los diferentes canales para preparar las imágenes para hacer la segmentación de la formas mas fácil y precisa.

9.2.1. Transformación Gamma

En primer lugar, realizarán un pre-procesamiento utilizando la transformación gamma. Para esto deben realizar la transformación usando dos gammas mayores a uno y dos menores que uno. Ya que la segmentación la harán por medio de umbralización, es importante realizar un análisis por medio de sus histogramas. Para esto deben calcular el histograma tanto de la imagen original como de las imágenes con cada una de las transformaciones. Incluyan en su informe un subplot como el de la Figura 6 por cada canal de la imagen donde se observen tanto las transformaciones, el canal original y sus histogramas de intensidades. En su informe responda:

- ¿Cuál es el objetivo que desean lograr haciendo estas transformaciones?
- ¿Cuál es la diferencia entre usar un gamma mayor a 1 y uno menor a 1?
- ¿Cómo alteró la transformación gamma el histograma de cada canal de color?
- ¿Cómo es el histograma que desearían obtener?

- ¿Qué gamma obtuvo un resultado que ustedes consideran mejor para cada canal? Justifiquen su elección para cada uno de los canales y mantengan esta transformación de cada canal para continuar con el desarrollo del miniproyecto.

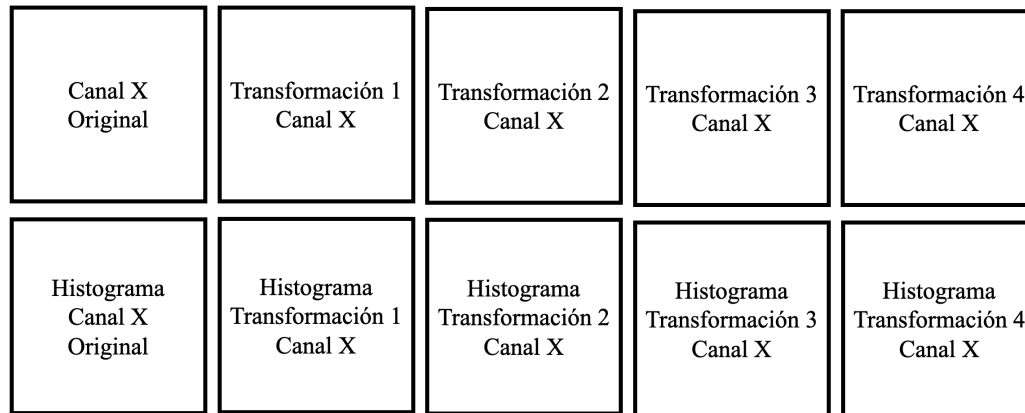


Figura 6: Ejemplo del subplot de las transformaciones de intensidad y sus histogramas.

9.2.2. Ecualización-Especificación

Otra alternativa que se podría llevar a cabo es transformar la imagen por medio de especificación o ecualización. Aunque en esta ocasión no lo implementarán, en su informe responda:

- ¿Consideran que la ecualización sería útil para esta tarea? ¿Por qué?
- ¿Considera que la especificación sería útil para esta tarea? ¿Por qué?
- En el caso de considerar la especificación como útil, mencione qué tipo de imagen debería usarse (qué tipo de histograma debería tener y cómo se vería esta imagen).

9.3. Experimentación

En esta sección harán la experimentación usando cada una de las combinaciones entre los métodos de umbralización (para segmentar) y los métodos de post-procesamiento para eliminar los artefactos. Una vez implementadas las dos primeras funciones harán una general que realizará el procesamiento completo de una imagen. Esta deberá ser capaz de realizar todo el procesamiento de la imagen de entrada y retornar las predicciones de detección de la imagen.

9.3.1. Segmentación

Para hacer el paso de la imagen a una mascara de segmentación van a usar la umbralización por medio de 3 métodos diferentes, es decir que van a utilizar 3 métodos diferentes para obtener umbrales y así obtener una imagen binaria. Para esto implementen una función que realice la umbralización por estos 3 métodos. Estos serán por medio del umbral de otsu, un percentil que ustedes determinarán y dos umbrales determinados arbitrariamente por ustedes que deberán justificar en su informe.

Nota: En caso en que consideren que más de un canal de color tiene información útil para generar la segmentación final, pueden realizar una o varias operaciones (i.e. operaciones booleanas) entre las máscaras de estos canales.

En su informe justifiquen la elección del canal o canales (junto con las operaciones) de interés que usaron para obtener la segmentación final.

Esta función internamente deberá hacer la selección del canal o los canales que vayan a usar, al igual que la operación binaria a utilizar en caso de requerirlo. Esta deberá tener la siguiente estructura:

Sintaxis:

```
thresholding_Codigo1_Codigo2(img, thersh_method)
```

Parámetros de entrada:

- `img`: Imagen de interés (con sus 3 canales).
- `thersh_method`: Metodo de umbralización que se va a usar. Puede ser “otsu”, “percentile” o “double”.

Respuesta:

- `mask`: binarización de la imagen por medio del método que se usó en la entrada.

9.3.2. Post-Procesamiento

Habiendo realizado la segmentación, es posible que existan componentes conexos que no tienen relación alguna con el objeto que se está queriendo segmentar. Para esto es necesario que se realice un post-procesamiento para poder eliminar estos artefactos de la imagen. Por ello, ahora harán el post-procesamiento necesario para así poder tener únicamente los elementos conexos que pertenezcan a los glóbulos blancos.

Componentes conexos

Inicialmente realizarán la limpieza de la segmentación mediante el método de componentes conexos. Aquí deben usar la información de aquellas agrupaciones de píxeles y su tamaño para poder determinar cuales pertenecen a la segmentación deseada y cuales son artefactos o ruido. Para esto deben discutir la metodología planteada por cada uno en la pre-entrega en su informe para poder determinar el algoritmo que implementarán.

Sub-muestreo/Re-muestreo de la imagen

Otra forma en la que se pueden eliminar aquellos artefactos de menor tamaño es bajando la resolución al punto de que sean eliminados y mas adelante restaurar la imagen sub-muestreada a su tamaño original. Para este procedimiento pueden utilizar funciones de diferentes librerías como `rescale` de `scikit-image` para lograr alterar el tamaño de la imagen. Adicionalmente, en su informe responda:

- ¿Qué tamaño tenía su imagen sub-muestreada?
- ¿Qué interpolación usó para restaurar al tamaño original?

- ¿Con todas las interpolaciones la imagen continua siendo binaria?
- ¿Qué se puede realizar en caso de que la imagen deje de ser binaria al ser interpolada?

Implementación

Una vez realizada la discusión deben implementar su algoritmo en la siguiente función donde el procesamiento por medio de componentes conexos se realice al introducir el parámetro `post_proc` con el valor “component” mientras que debe realizar el proceso por el metodo de sub-muestreo/re-muestreo al introducir “scale” en dicho parámetro.

Sintaxis:

```
postprocesing_Codigo1_Codigo2(im_file, post_proc)
```

Parámetros de entrada:

- `img`: Imagen binaria sobre la cual deseen hacer post procesamiento.
- `post_proc`: Metodo de post procesamiento que se va a usar. Puede ser “component” o “scale”.

Respuesta:

- `mask`: Imagen binaria procesada sin los componentes de menor tamaño.

9.3.3. Función de procesamiento

Habiendo implementado las diferentes alternativas para el procesamiento, implementen una función para realizar de forma iterativa el procesamiento de todas las imágenes. Esta debe abrir cada imagen, realizar el pre-procesamiento y segmentación que definieron, y finalmente debe aplicar el post-procesamiento y usar la función que realizaron la entrega pasada para obtener las predicciones de detección. Esta debe tener la siguiente estructura:

Sintaxis:

```
Procesing_Codigo1_Codigo2(im_file, img_id, thersh_method, post_proc)
```

Parámetros de entrada:

- `im_file`: Ruta en la cual se encuentra la imagen original que se desea procesar.
- `img_id`: Id de la imagen que se desea procesar.
- `thersh_method`: Metodo de umbralización que se va a usar. Puede ser “otsu”, “percentile” o “double”.
- `post_proc`: Metodo de post procesamiento que se va a usar. Puede ser “component” o “scale”.

Respuesta:

- `detection`: Lista de diccionarios con las predicciones.
- `segmentation`: Imagen binaria previa al post procesamiento.
- `clean_segmentation`: Imagen binaria procesada sin los componentes de menor tamaño.

Para determinar su *score* utilizarán la función `pred_score(img)` que se encuentra en el archivo `utils.py` dentro de la carpeta `data_mp1`, a la cual deben pasarle como único parámetro un recorte en el área de la predicción de la imagen original(con sus 3 canales).

9.3.4. Validación

Utilizando las funciones implementadas, realicen experimentos donde obtengan la métrica de interés variando el método de umbralización y el método de post procesamiento en su *split* de validación. Aquí deben realizar e incluir en su informe una tabla que contenga los 6 experimentos diferentes con su respectiva área bajo la curva. Para esto deben realizar una tabla como la del Cuadro 2, donde en negrilla resalten su mejor combinación de métodos.

Cuadro 2: Ejemplo de tabla de experimentación

Umbralización	Post-Procesamiento	Area bajo la curva (mAP)
Otsu	Componentes	
	Re-escalamiento	
Percentil	Componentes	
	Re-escalamiento	
Doble Umbral	Componentes	
	Re-escalamiento	

Adicionalmente, en su informe incluyan un subplot como el de la Figura 7 con sus resultados cualitativos de la experimentación.

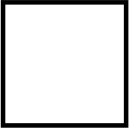
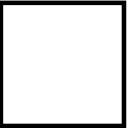
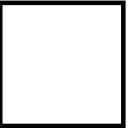

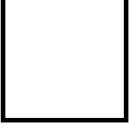
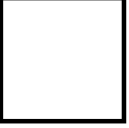
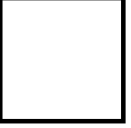
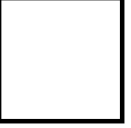
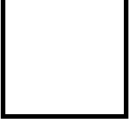
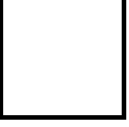
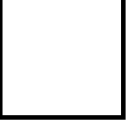

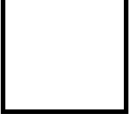
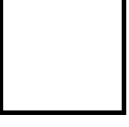
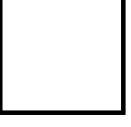

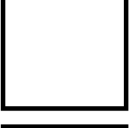

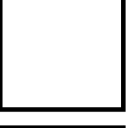

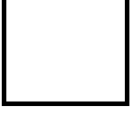
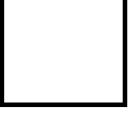
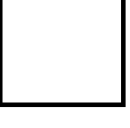
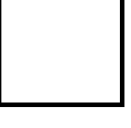
	Imagen original	Máscara original	Máscara procesada	Imagen+ detecciones
Exp. 1				
Exp. 2				
Exp. 3				
Exp. 4				
Exp. 5				
Exp. 6				

Figura 7: Ejemplo de subplot del proceso de experimentación.

Por último, en el informe respondan:

- ¿Cuál combinación les dio un mejor resultado cuantitativamente? ¿Por qué creen que esta combinación les dio mejores resultados?
- ¿Sus resultados cualitativos son congruentes con los cuantitativos?
- ¿Qué errores o fallas pueden observar en los resultados cualitativos?
- ¿Qué parte del procesamiento genera más errores? ¿Por qué?

9.4. Prueba

Finalmente, después de elegir la combinación de método de segmentación y post-procesamiento de su método final, realizarán la ultima etapa de esta entrega y este miniproyecto: la evaluación. En esta sección harán uso de la partición (*split*) de prueba para así poder evaluar de forma objetiva el desempeño de su algoritmo. Para esto realicen el procedimiento usando su método final sobre el conjunto de test. Realicen la curva de precisión y cobertura para los 3 índices de Jaccard que usaron en la entrega anterior. Además de esto realicen un subplot como el de la Figura 8 donde se vean 5 imágenes diferentes, las anotaciones y sus predicciones.

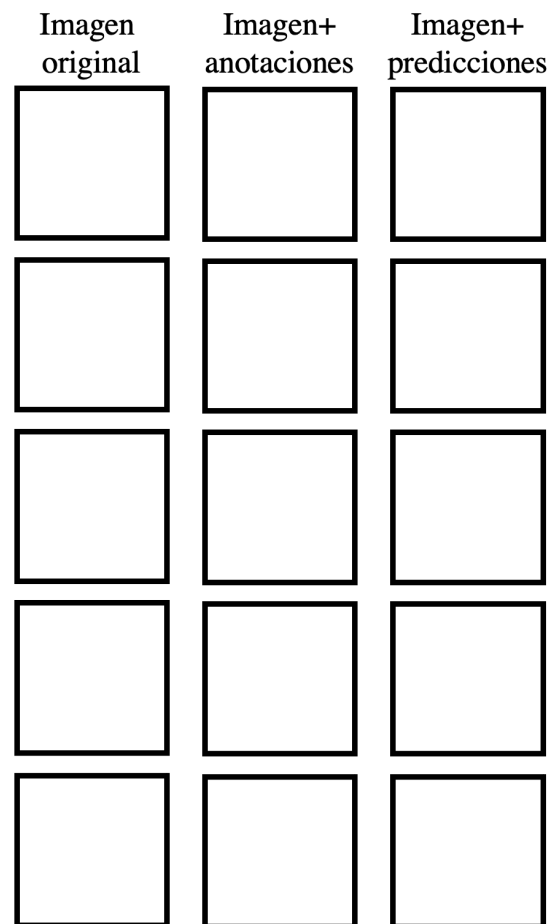


Figura 8: Ejemplo de subplot de los resultados cualitativos en *test*.

En su informe responda:

- ¿Cómo considera que fue la detección con el método final que realizó?
- ¿Los resultados en validación y prueba son congruentes? ¿Cómo se relacionan?
- ¿Qué aspectos consideran que podrían mejorar? ¿Cómo los mejoraría?