1. **Introducción**

En los últimos siglos los grandes avances tecnológicos, medicinales y sociales han llevado a que la humanidad haya crecido exponencialmente, alcanzando a la fecha de hoy una población global de más de 7.5 miles de millones de personas. Esto ha llevado consigo una deforestación masiva con el fin del uso del área deforestada para la agricultura, urbanización o para construir diferentes proyectos de infraestructura. Esto ha tenido un efecto en el clima de la Tierra y en la habilidad de miles de especies para subsistir. Es por esto mismo, que uno de los principales enfoques de los movimientos de conservación y protección ambiental en los últimos años ha sido la protección de bosques y junglas por medio del monitoreo de estos.

Previamente una de las técnicas más populares para el monitoreo de bosques ha sido por el uso de grupos voluntarios para el monitoreo de áreas específicas. Algunos de los métodos utilizados para el monitoreo como *20-trees count*, *Bitterlich gauge method* y *Disturbance Checklist transect*, se enfocan en contar la cantidad de árboles de una especie en un área específica y registrar diferentes perturbaciones alarmantes en dicha zona. Se ha estimado que utilizando estos métodos para bosques estadunidenses conlleva un costo entre $0.04 a $0.12 por hectárea cubierta (Holck, 2008). Sin embargo, conforme el área es más remota y entre mayor sea la expansión de área, los costos tienden a subir. Esto a evitado que se puede monitorear extensas regiones utilizando estos métodos tradicionales, ya que estos pueden tener grandes costos para la movilización de voluntarios a zonas lejanas.

Una de las nuevas tendencias en los últimos años a sido la utilización de fotografías satelitales para monitorear los bosques. Esto permite extraer una gran cantidad de información sobre una extensa región de manera rápida, efectiva en costos y fácilmente expandible. Sin embargo, debido a la gran cantidad de datos es necesarios utilizar modelos computacionales para lograr clasificar toda la data obtenida. Es por esto por lo que grandes fundaciones de conservación como Sustania han invertido en modelos de Deep Learning que permitan detectar áreas deforestadas, los cuales permiten que se enfrente el problema de la conservación natural de una manera más eficiente (Pedersen, 2019).

Es por todo lo que se ha mencionado previamente, por lo cual, este trabajo tiene como finalidad la investigación del monitoreo forestal por medio de imágenes satelitales y modelos computacionales. Esto por medio de la formación de un modelo de Deep Learning para la clasificación de imágenes satelitales. Lo cual permitirá estudiar la efectividad del modelo creado, así como para obtener conclusiones sobre las capacidades, beneficios y desventajas de dichos modelos.

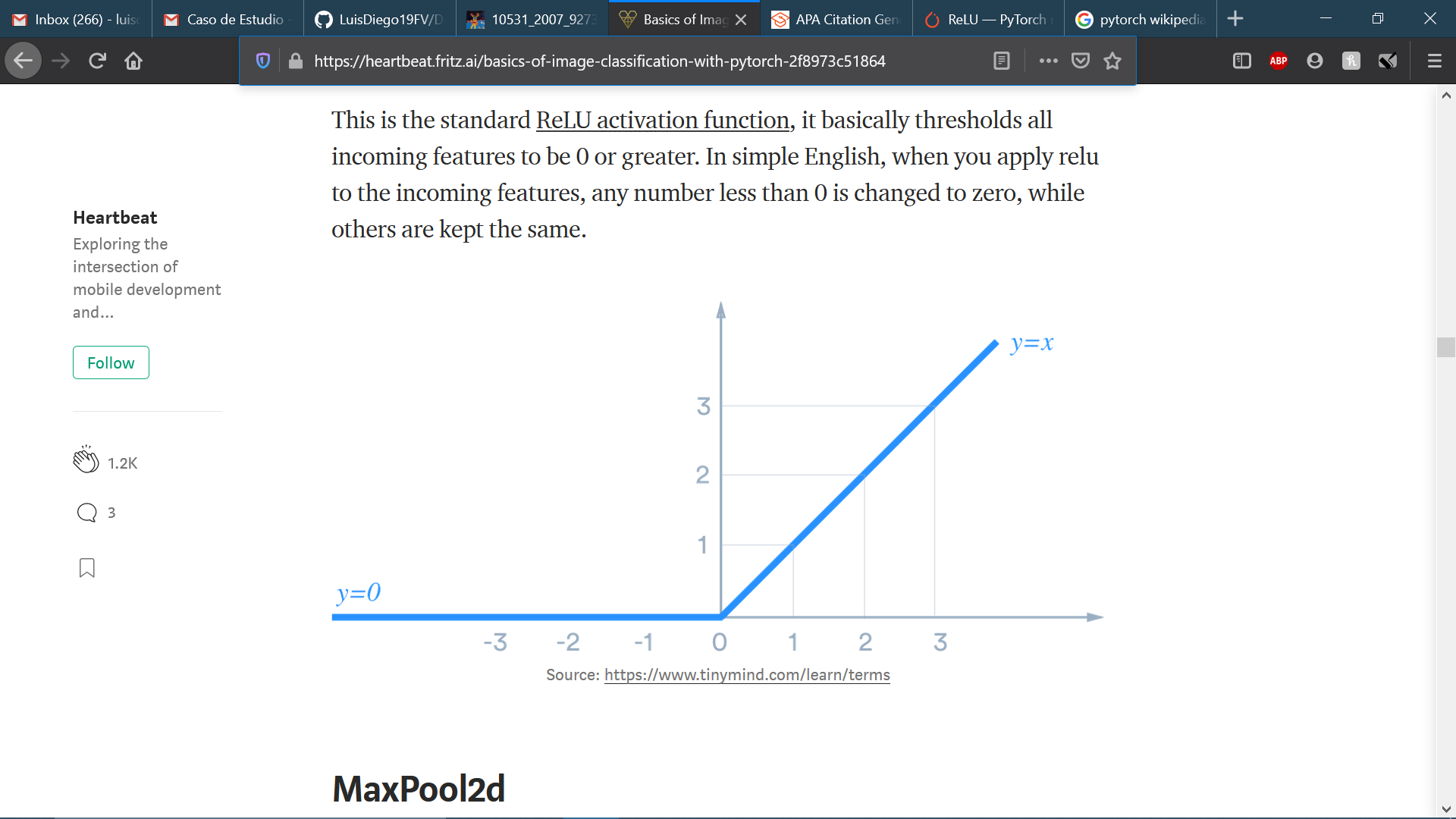
1. **Marco teórico**

El reconocimiento de imágenes por medio de modelos computacionales funciona mediante tres principios: que todas las imágenes pueden ser representadas bajo un array de bits que representan los colores de cada píxel, que se puede encontrar patrones bajo el análisis de grupos de pixeles y que los modelos solamente deben de enfocarse en los factores distintos de las imágenes más que en los similares. Bajo estos tres principios se es capaz de entrenar modelos computacionales los cuales se enfoquen en diferentes características de cada imagen para poderlas clasificar adecuadamente.

El pipeline de un modelo de Deep Learning (DL) se divide en tres partes: función de activación, reducción de data y entrenamiento de neuronas (Olafenwa, 2020). La función de activación acentúa todas las características de una imagen al asignarles un valor numérico dentro de un rango asignado. La reducción de la data es una función utilizada para reducir la cantidad de variables que utilizará el modelo, permitiendo que se utilice menos poder de procesamiento en la creación y entrenamiento del modelo computacional. Finalmente, las neuronas es la red de nodos que se forman para darle valor a diferentes componentes de la imagen basado en un peso que se establece en el entrenamiento basado en un dataset de entrenamiento.

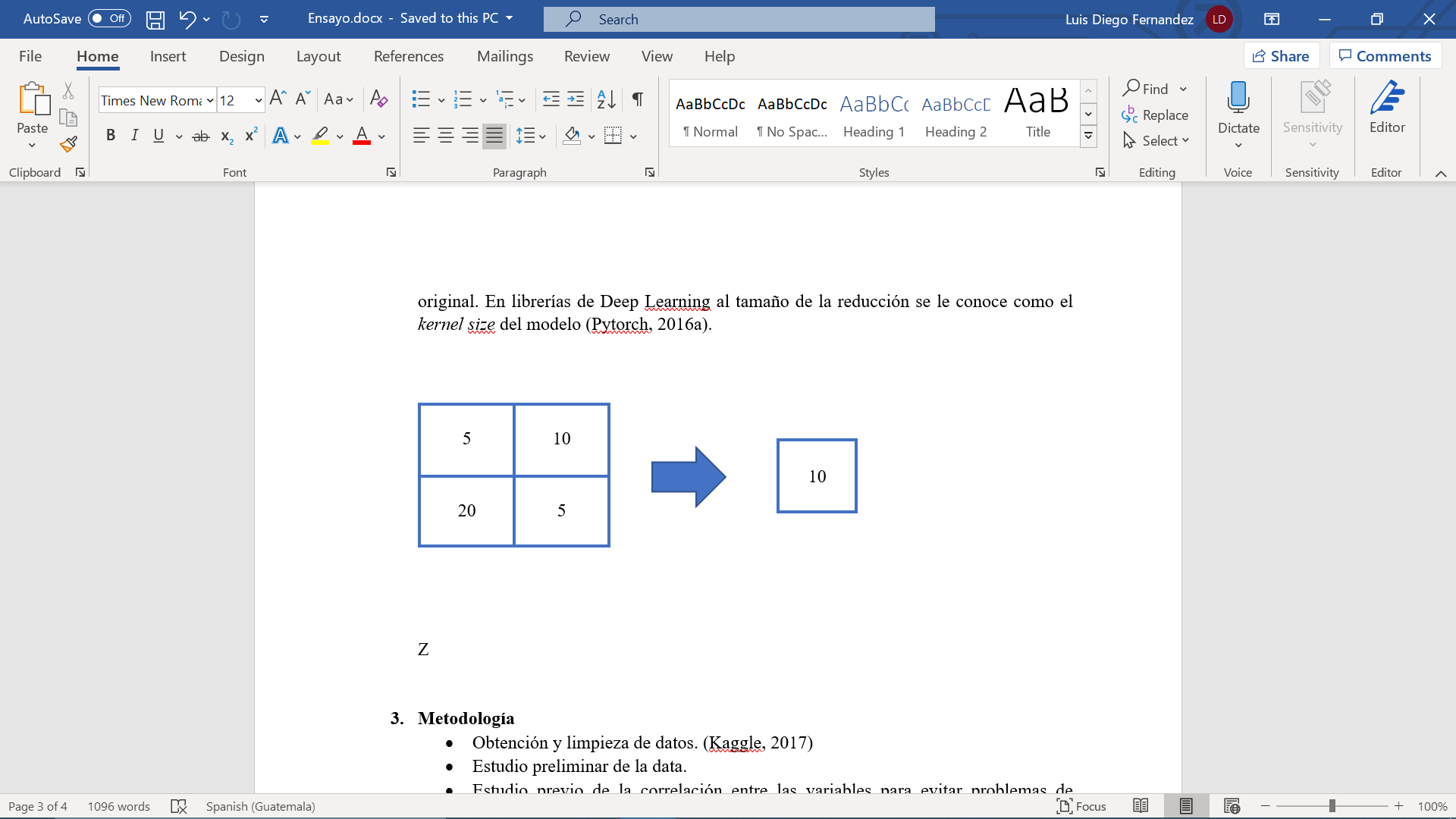
La función de activación que se utilizará en este trabajo será ReLU (Pytorch, 2016b). Esta función de activación permite transformar cualquier valor menor a cero a cero, mientras que, los valores mayores a cero mantienen su valor original. Esto puede modificarse para utilizar un valor diferente a cero como el pivote. Este será utilizado para acentuar en la data únicamente aquella información necesaria, desechando de esta manera parte del ruido de la imagen.

*Figura 1: Ejemplo de función de activación ReLU (Olafenwa, 2020)*



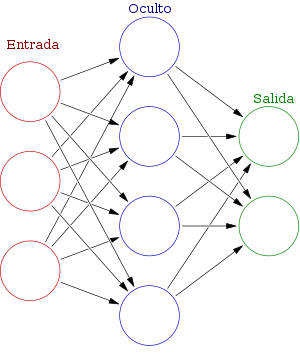
La siguiente parte luego de aplicar la función de activación es utilizar una función de reducción. Como se menciono previamente la función de reducción se utiliza para reducir la cantidad de información provista por data cruda. En imágenes esto se hace por medio de una reducción de pixeles. Esto por medio de promediar el valor de un grupo de pixeles en uno solo dato. Por ejemplo, si se tiene una imagen de 16x16 y se hace una reducción de 4, se obtendría una imagen 4x4 que contiene los valores promedios de los pixeles de la imagen original. En librerías de Deep Learning al tamaño de la reducción se le conoce como el *kernel size* del modelo (Pytorch, 2016a).

*Figura 2: Ejemplo de función de reducción*



La ultima sección del pipeline de Deep Learning es la red neuronal. Esta sección se encarga de formar una red de nodo que la asigna un peso a cada transición cuando se va a entrenando el modelo. Esto funciona ya que al entrenar el modelo se le da información sobre el objeto que se esta identificando en la imagen. Esto permite que el modelo neuronal se adapte a los datos dados y se logre clasificar en el futuro imágenes similares.

*Figura 3: Ejemplo de estructura de una red neuronal*



1. **Metodología**

Metodología en grandes rasgos:

El modelo para utilizar es un modelo creado en Python 3.8 utilizando principalmente la librería de DL de PyTorch para la creación de un modelo de clasificación de imágenes forestales. Este será entrenado utilizando imágenes del Amazonas las cuales han sido clasificadas manualmente de manera previa. Finalmente, esto será utilizado para analizar la efectividad del modelo y utilizarlo con imágenes fuera del dataset de entrenamiento.

Metodología, pasos específicos:

* Obtención de la data:

La data se obtuvo de la competencia lanzada por Kaggle: *Understanding the Amazon from Space* del 2017 (Kaggle, 2017), la cual sigue habilitada para la obtención en la data en el URL: https://www.kaggle.com/c/planet-understanding-the-amazon-from-space/data.

* Estudio preliminar:

En este se busca encontrar diferentes aspectos de los datos que pueden servir para la formación del modelo. La información para extraer es la siguiente:

* Las diferentes clasificaciones que se le puede a las imágenes.
* Los diferentes componentes que puedan causar ruido dentro de las imágenes,
* Los tamaños promedios de las imágenes.
* Obtener ejemplos generales de cada clasificación posible para las imágenes.
* Creación del modelo:

Utilizando Python 3.8 y las librerías de Pandas, csv y PyTorch se realizará las siguientes partes del modelo:

* Definir la clase de nodos.
* Definir la función regresiva.
* Definir la función progresiva.
* Definir la función de activación.
* Definir la función de reducción.
* Definir los parámetros del modelo neuronal.
* Definir la función de entrenamiento.
* Entrenamiento del modelo:

Se utiliza la data obtenida en conjunto a la función de entrenamiento para crear el modelo de DL.

* Análisis:

En esta sección se busca obtener cuantas fotografías utilizadas para el entrenamiento es capaz de clasificar correctamente el modelo creado.

1. **Resultados**

En esta sección se mostrarán gráficas realizadas la sección anterior. En estas se piensa mostrar:

* La exactitud del modelo realizado.
* La exactitud del modelo por cada grupo de tipo de imagen en el dataset.

1. **Discusión**
2. **Conclusiones**
3. **Bibliografía**

Alake, R. (2020, June 8). (In-depth) Machine Learning Image Classification With TensorFlow. Retrieved August 8, 2020, from https://towardsdatascience.com/in-depth-machine-learning-image-classification-with-tensorflow-2-0-a76526b32af8

Bronstein, C. (2019, February 25). Land-Use and Deforestation in the Brazilian Amazon - Towards Data Science. Retrieved August 8, 2020, from https://towardsdatascience.com/land-use-and-deforestation-in-the-brazilian-amazon-5467e88933b

Holck, M. H. (2008). Participatory forest monitoring: an assessment of the accuracy of simple cost–effective methods. Biodiversity and Conservation, 17(8), 2023-2036.

Kaggle. (2017, February 20). Planet: Understanding the Amazon from Space | Kaggle. Retrieved August 8, 2020, from https://www.kaggle.com/c/planet-understanding-the-amazon-from-space

Olafenwa, J. (2020, January 29). Basics of Image Classification with PyTorch - Heartbeat. Retrieved September 13, 2020, from https://heartbeat.fritz.ai/basics-of-image-classification-with-pytorch-2f8973c51864

Pedersen, R. S. (2019, June 11). Watching, listening, and learning: how AI can reduce deforestation. Retrieved September 12, 2020, from https://sustainiaworld.com/watching-listening-and-learning-how-ai-can-reduce-deforestation/

Pytorch. (2016a). AvgPool1d — PyTorch master documentation. Retrieved Sep. 13, 2020, from https://pytorch.org/docs/master/generated/torch.nn.AvgPool1d.html?highlight=kernel%20size

Pytorch. (2016b). ReLU — PyTorch master documentation. Retrieved September 13, 2020, from https://pytorch.org/docs/master/generated/torch.nn.ReLU.html#relu

Tensor. (n.d.). Image classification | TensorFlow Core. Retrieved August 8, 2020, from https://www.tensorflow.org/tutorials/images/classification