

Entrenamiento de una red neuronal convolucional utilizando un set de datos sintetico para reconocimiento de objetos

Jesus Eduardo Ortiz Sandoval

Departamento de electrónica

Universidad Tecnica Federico Santa Maria

Valparaiso, Chile

jesus.ortiz@sansano.usm.cl

Abstract—This paper is an development about a existing technique 3D modelling for building a dataset for image recognition using covolutional neural networks (Deep Learning), it's really interesting the results and the evaluation of the results since a dataset real and dataset sintetic.

Index Terms—vision, deep learning, convolutional, groundtruth, 3D models

I. INTRODUCCIÓN

En la actualidad el mayor problema del aprendizaje supervisado de mquinas, que ha demostrado ser un gran calificador con innumerables posibles aplicaciones en la sociedad, es la construccin del set de datos para entrenamiento y para pruebas. Si hablamos solamente en el ramo de la visin por computador y se quisiera hacer el reconocimiento de una hoja, o de un producto, se necesitaran por lo menos 10.000 imgenes con diferentes escenarios y caractersticas particulares para permitir un aprendizaje correcto y tener una taza de acierto aceptable en el reconocimiento de este objeto en entornos sin control. [1].

Si la construccin de estos sets de datos se hiciera manualmente, el tiempo que podra demorar el desarrollo de un solo proyecto sera muy largo y por ende no se podra tener resultados certeros o desarrollar mejoras de los proyectos. Utilizando un software de modelado, lo que se busca es generar estos sets de datos gigantes en un tiempo reducido, por ejemplo, de una lata de Coca Cola se pueden generar 16.000 imgenes con diferentes fondos, inclinaciones e incidencias de luz en aproximadamente una hora y media en una computadora con una potencia de computo media-alta. Esta diferencia de tiempo permite explorar mtiples opciones de desarrollo de proyectos y de productos, as como abre una incgnita muy interesante para la investigacin, la cual es la optimizacin de los algoritmos de clasificacin y seguimiento con set de datos artificiales pero que sean aplicados a situaciones reales. Ello permitira determinar con certeza cuantos datos son necesarios, de que tipo y en qu forma para que al aplicar los algoritmos se tenga una taza de acierto alta y sea eficiente computacionalmente.

Las aplicaciones a futuro que tiene este proyecto comercialmente son muchas, en el rea de ventas, para las cadenas

de distribucin de productos tener una herramienta de reconocimiento de productos para ofrecer servicios innovadoras es muy importante, Amazon Go es un ejemplo de que en los nuevos mercados es importante cubrir las necesidades del consumidor. En otras reas tambin se puede implementar y dar solucin a problemticas reales, en establecimientos donde est prohibido el uso del telfono celular, o en la industria minera cuando un operario se encuentra sin el equipamiento necesario, entre otras aplicaciones.

II. ESTADO DEL ARTE

En el rea de la visin artificial se encuentran en desarrollo y finalizados muchos proyectos que tienen como fin la deteccin, clasificacin, caracterizacin o extraccin de parmetros de diferentes objetos con una finalidad investigativa o con el nimo de lucro mediante proyectos que tengan impacto en la sociedad.

A. Trabajos actuales

En muchos proyectos una de los principales objetivos es el de investigar la eficiencia del uso de extraccin de patrones para clasificar diversidad de elementos usados en data sets [2]. En esta rama se encuentran situaciones individuales que solas son reas de conocimiento, una que es muy relevante hace referencia al data sets con el que se va a trabajar, cuando se tienen set de datos des balanceados causan desafos crticos creando clases negativas, clases positivas y problemas de decisin en las mquinas de aprendizaje [3]. Es importante optimizar la construccin del set de datos que sern destinados al entrenamiento de los clasificadores, para no desperdiciar tiempo computacional y en procura de tener una alta tasa de efectividad. En el trabajo de Piri [3], se hacen uso de herramientas computacionales de inteligencia artificial como lo son mquinas de soporte vectorial para hacer una optimizacin o arreglo de set de datos des balanceados utilizados en la minera de datos.

En la investigacin titulada Machine Learning for gravity spy: Glitch classification and data sets., explican adecuadamente lo que para ellos es la preparacin del set de datos, as como la relevancia para poder encontrar resultados adecuados,

y mencionan que una de las grandes dificultades es poder construir de una manera adecuada un set de datos que cumpla con todos los posibles niveles de complejidad necesarios para hacer clasificación de una manera asertiva [4], y se plantean 22 normas que deben ser cumplidas al momento de construir el set de datos.

En la sección de visión por computador, son muchos los proyectos que se han desarrollado, y el aprendizaje de máquinas ha demostrado ser de mucha utilidad cuando se quieren reconocer características nicas o propias de los objetos, como color, tamaño o texturas, en el trabajo titulado Gaussian derivative models and ensemble extreme Learning machine for texture image classification se habla de la importancia de la clasificación de imágenes, como un tema importante del procesamiento de imágenes, donde cada pixel puede tener una o diferentes clases dependientes de la textura de los objetos, y en donde la clasificación de estas texturas juega un papel fundamental en la identificación de productos para las industrias [5].

Con el uso del Deep Learning en el aprendizaje de computadores se ha buscado emular el comportamiento humano para optimizar los procesos en la inteligencia artificial. En muchas de las investigaciones los profesionales siempre han querido crear un modelo que intente replicar completamente el sentido de la visión del ser humano, construyendo una solución multi-dimensional donde no solo la imagen sea el centro del estudio, también otras variables, como luz, distancia etc [6]. Al tener sistemas multivariados, es esencial el procesamiento correcto de la información para poder tener resultados relevantes.

Muchos estudios se han centrado en la limitación del manejo de grandes datos para el aprendizaje de sistemas supervisados, cuando los datos son inciertos, o no se tiene suficiente información del proceso se hace necesario la implementación de un aprendizaje automático que cumpla con la convergencia de hipótesis y de comportamiento de un proceso [7]. En un escenario con la creación de familias automáticas de aprendizaje se puede implementar un algoritmo que recibiendo algunas indicaciones de entrada se tenga un estructura automática de entrada-salida de un sistema generando la adaptabilidad que es necesaria en muchos proyectos actuales [8].

El Deep Learning constituye una técnica moderna para el procesamiento de imágenes y análisis de datos, con resultados prometedores y un gran potencial en diferentes ramos como lo son la salud, agricultura, producción de alimentos, transporte, seguridad entre otros [9]. Es importante resaltar que actualmente ya un gran número de industrias están remplazando la clasificación visual humana, por computadoras que permitan una relación costo-beneficio mayor, en esta área el Deep Learning demuestra tener grandes aplicaciones pues comparado con los clasificadores normales, las redes

neuronales convolutivas, no solo tienen altos porcentajes de acierto, también permiten extraer características y patrones para otro tipo de análisis construyendo escenarios de alta complejidad. [10] [11].

III. DATOS SINTÉTICOS Y GAN

En la red podemos encontrar muchísimos tipos de algoritmos para reconocimiento de objetos, algunos de los más interesantes fueron nombrados en la sección anterior, la novedad de este trabajo radica principalmente en el set de datos para entrenar la red neuronal. Mucho se habla del álgebra tensorial y de la necesidad de tener cientos de gigas de información así como potentes computadoras que permitan procesar grandes cantidades de datos, así se estudia la BigData o DataScience, pero ese es el principal problema de todo algoritmo, los datos, en bases de datos científicas podemos encontrar millones de datos, etiquetados, registrados que nos permiten entrenar un clasificador bueno, pero si quisieramos tener una aplicación real que genere ingresos debemos recurrir a construir este set de datos por cuenta propia, si quisieramos reconocer todos los elementos de un supermercado, tal como en este momento lo hace Amazon Go con su proyecto de tienda sin cajeros en Seattle, EEUU, necesitaríamos tomar millones de imágenes a los miles de productos que se encuentran, esto no es práctico y suele ser uno de los grandes inconvenientes para aplicar y desarrollar este tipo de estrategias.

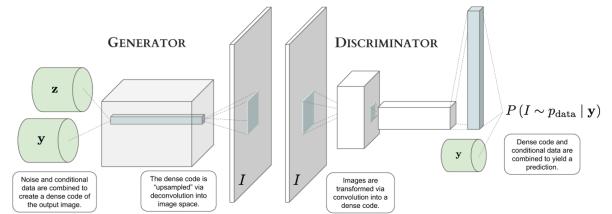


Fig. 1. Estructura de una red GAN

A. GAN

Las redes generativas antagónicas (Generative Adversarial Network) fueron introducidas por Ian Goodfellow en el año 2014, y según expertos en AI es la mejor idea que ha surgido en los últimos 10 años. La teoría nos indica que este modelo posee dos redes, una discriminatoria y una generativa, el objetivo de cada una es hacer fallar lo máximo posible a la otra [12], red generativa debe ser capaz de producir una imagen o un dato que al ser evaluado por el modelo discriminatorio la probabilidad de que la red acierte sea

$$D = \frac{1}{2} \quad (1)$$

esto simplemente nos indica que es tan bueno el generador que su función discriminativa es completamente aleatoria. Existe un gran número de parámetros para el entrenamiento de una red neuronal profunda, sin embargo nunca son suficientes debido a la estructura de estos algoritmos y para poder encontrar resultados aceptables se debe recurrir a dejar a un lado

características que en algún punto puede ser relevantes [13]. Este tipo de redes tiene una gran respuesta para problemas de segmentación en imágenes, así como en desafíos de super alta resolución [12].

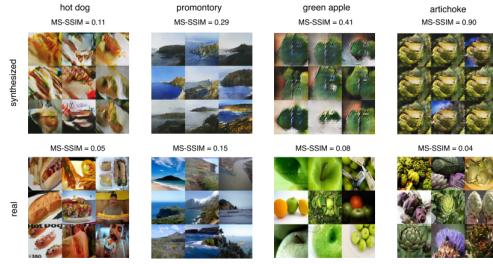


Fig. 2. Imágenes generadas vs reales

En la figura 1 podemos encontrar el resultado de una implementación de una variante de GAN, donde el modelo generativo con base a la información recibida genera unas figuras que fácilmente podrían ser clasificadas como datos reales, en la actualidad se encuentran demasiadas versiones de este tipo de redes ACGAN, SGAN, WGAN, TGAN, CGAN, etc, en cada modelo encontramos diferentes tipos de función de activación o modulos que se añaden para tener mejor rendimiento o una aplicación específica.

B. Datos sintéticos

Esta idea fue desarrollada en la Universidad de Massachusetts utilizando una herramienta que se llama ADOBE 3DS MAX, que tiene licencia gratuita para los estudiantes, este entorno de desarrollo permite hacer render 3D, diseños, modelos y correr scripts, en estas porciones de código podemos permitirnos con un modelo 3D, fondos y texturas generar imágenes en JPEG de 3 capas, también se puede manejar el formato PNG con la cuarta capa de transparencia.



Fig. 3. Imagen real y renderizada [?]

Los sets de datos son muy importantes en la investigación de visión por computador, cada vez con el crecimiento exponencial de la ley de Moore's se necesitan grandes cantidades de datos, los modelos 3D gratuitos en estos momentos inundan internet, a parte de también ser fácilmente implementables, en la figura 2 encontramos una imagen real y otra generada en la computadora, hablamos de set de datos sintético al poder obtenerlo todo virtualmente, sin necesidad de tomar una fotografía del mundo real, generando nuestras imágenes a partir de un modelado 3D.

IV. METODOLOGIA

La metodología del proyecto la podemos resumir en tres estaciones, modelamiento, groundtruth(set de datos) y la red

neuronal que incluye entrenamiento del modelo, validación y evaluación del algoritmo.

A. Modelamiento

Para hacer el modelamiento de los objetos se va a utilizar el software ADOBE MAX 3Ds, cuya imagen de presentación se observa en la figura 3, para comenzar con el desarrollo del proyecto se plantean varios tipos de productos a utilizar, botellas de vino, desodorantes, alimentos congelados, pero surgen unos problemas respecto a las etiquetas, tomando este referente se decide empezar a hacer el set de datos con envases de papas pringles de tres sabores **QUESO**, **CEBOLLA** y **CLASICAS**. La idea es hacer un diseño general con solo la textura de la tapa y luego aplicar diversas texturas al envase para generar el set de datos.

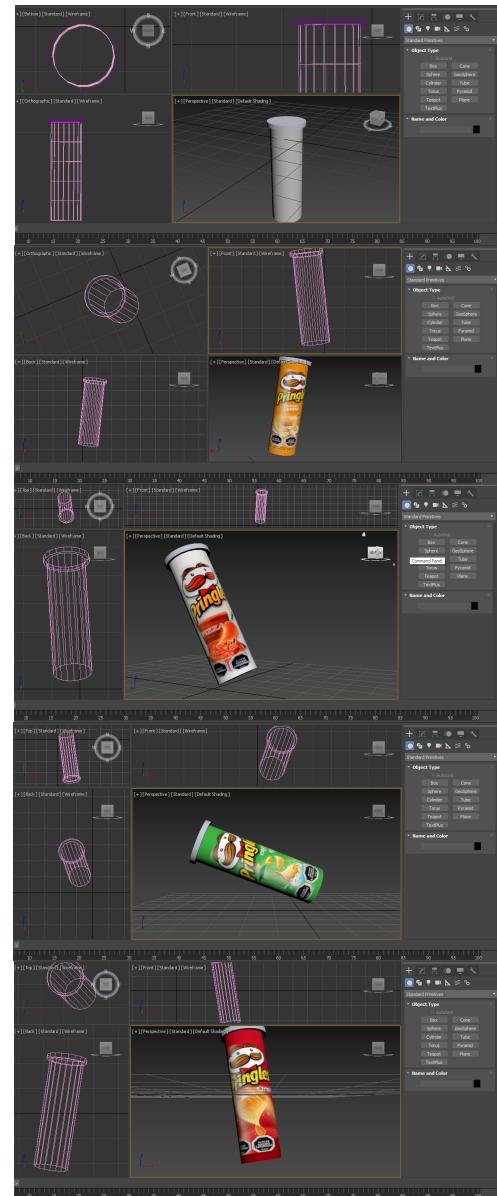


Fig. 4. Render 3D pringles

Una vez que se tienen las texturas correctamente armadas y verificadas se diseña una estructura de textura para aplicar al modelo, una vez que se hace este paso, se hace la unión de los elementos a través de la herramienta ProBoolean, para poder hacer este paso siempre es necesario que la textura se aplique de forma individual, si se omite este paso, el resultado de la textura es incorrecto y genera errores en el diseño. A continuación en la figura 4 están los diseños finalizados de las 4 clases posibles para el entrenamiento.

B. Groundtruth

Para generar el groundtruth de las clases correspondientes y el set de datos para el entrenamiento de la red neuronal con objetivo de identificar y clasificar las papas pringles con 3 clases, una vez que se ha terminado el modelo se procede a la parte de programación que permite obtener el dataset correspondiente para poder implementar el entrenamiento del modelo neuronal. Como primer elemento se elabora un script para el programa ADOBE 3DS MAX que permite la renderización de los archivos, en esta sección de código se definen los parámetros necesarios para la generación de los datos, número de poses, iluminación de la escena, rotación del elemento y el fondo que se va a aplicar al renderizado. A diferencia de lo que se realiza normalmente, se encontró un set de fondos de supermercados y productos superior a 3k de imágenes, estos se utilizaron para generar las imágenes de entrenamiento, también del mismo tamaño, con los mismos parámetros se genera una imagen con fondo blanco, esta es una herramienta para el algoritmo de etiquetado y extracción de parámetros para el groundtruth adecuado. Se producen 25.000 imágenes de cada clase, la computadora en la que se ejecuta el algoritmo toma un tiempo aproximado de tres horas en generar las 25.000 imágenes.

blanco y en segunda opción con los mismos parámetros de orientación, iluminación, ubicación pero con unos de los fondos aleatoriamente escogidos. Con las imágenes generadas se escribe un algoritmo en octave que utilizando homografía, transformación de planos con la ayuda de las imágenes en fondo blanco se extraen las siguientes características:

$$\text{Class}, \text{Width}, \text{Height}, \text{Xmin}, \text{Ymin}, \text{Xmax}, \text{Ymax} \quad (2)$$

Estos elementos son exportados en un archivo .txt para cada elemento jpeg, es necesario tomar un paso más de pre-procesamiento y es la construcción de los archivos .csv, y de la distribución de los datos para entrenamiento y para testear. Es importante distribuir los datos en carpetas de train y test para generar los archivos record correspondientes, se tienen los 50.000 .txt equivalentes a cada imagen, con fondo blanco y fondo real, con algoritmos construidos en python se hacen dos procesos.

Construcción de datos para test y train, lo ideal es tomar aproximadamente 30 % de los datos para las carpetas test y train, al ser un número muy grande de datos y si se quiere también sacar archivos aleatorios un script que recorre la carpeta y va copiando archivos a los directorios correspondientes es necesario. Luego con las carpetas test, train y labels listas con los elementos se estructura un programa que crea los archivos csv, estos deben tener una estructura fija que solicita la red convolucional y es incluir en orden los siguientes parámetros **filename, width, height, class, xmin, ymin, xmax, ymax**, esto es necesario para poder ejecutar el script tfrecords, que construye el set de datos completo que es el que se va a ejecutar en el algoritmo correspondiente, este archivo de datos contiene unos tensores de altas dimensiones y es parte fundamental para lograr tener un entrenamiento exitoso.



Fig. 5. Modelos de clases con texturas

En la figura 5 observamos el resultado del modelamiento, en la parte superior el modelo pringles de cebolla con fondo

C. Segmentación y Algoritmo

Existen muchas estructuras de machine learning para hacer reconocimiento de objetos, mobilenet, inception, resnet, etc.

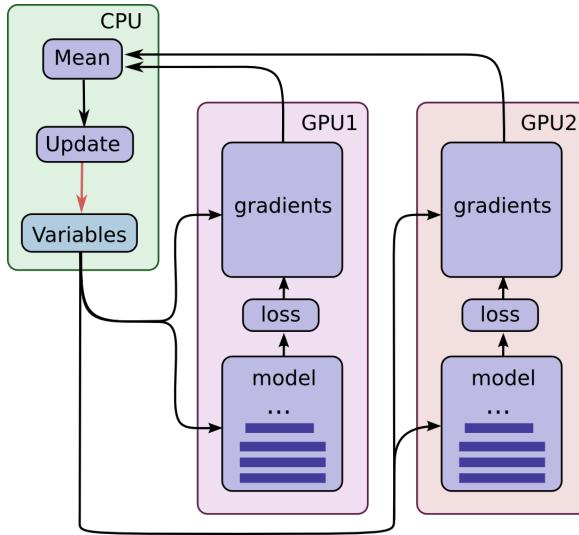


Fig. 6. Modelos inception

Para este proyecto se elige el siguiente modelo **faster rcnn inception v2 coco**, ya que en los backgrounds correspondientes se hacen excelentes comentarios sobre el rendimientos y la estructura de la red convolucional, se configuran los siguientes parametros:

Clases=cebolla, queso, clasica

Schedule= 9000 y 12000

Steps =20.000

Minimum loss= 0.2

Con los parametros establecidos se comienza el entrenamiento del modelo, la computadora en la que se ejecuta tiene una GPU de 2GB y 8GB de memoria RAM, se tiene un tiempo aproximado de 54 segundos para cada paso de entrenamiento, para alcanzar el objetivo de minima perdida se dejo haciendo iteraciones por aproximadamente 11 horas.

RESULTADOS

En este banchmarking se debe resaltar que el principal resultado es encontrar si la propuesta es viable, en el cronograma de trabajo junto con los objetivos del proyecto se marca que para este punto del proyecto el objetivo principal es poder determinar si con el set de datos sintetico se puede entrenar un modelo neuronal que logre diferenciar elementos reales utilizando vision por computador. El modelo que se obtiene como resultado despues de 1000 pasos de entrenamiento tiene una tasa promedio de perdida de 0.5 en todos los ultimos modelos, aunque es un numero bastante aceptable, se desea poder mejorar este parametro ya que un excelente clasificador debe estar en el orden de 0.2. En la figura 7 se observan algunas de

TABLE I
TABLA DE RESULTADOS

Número muestras	Resultados algoritmos		
	Evaluacion imagen	Evaluacion video	Camara
300	92%	70%	82%

Resultados resumidos.

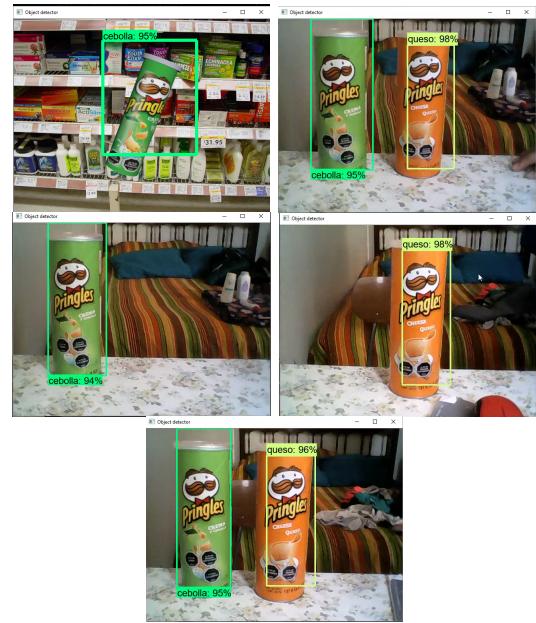


Fig. 7. Resultados del algoritmo

las pruebas que se le efectuaron al modelo entrenado, es de resaltar que el ambiente en el que se realizaron a las pruebas con la camara del notebook fue un ambiente no controlado, esto para verificar la robustes del algoritmo, un elemento que se advierte en todas las imagenes resultantes es un error en el boundbox del elemento y es elemento crucial para intentar mejorar en el transcurso del proyecto, como primera medida se debe poder determinar en que radica el fallo, para poder corregirlo y optimizarlo. Es necesario mejorar el algoritmo de segementación de la red neuronal, puede ser con el uso de super pixeles o el uso de un batch mayor, aunque para poder implementar esto es necesario contar con una GPU mucho mas poderosa. La clasificación promedio de las diferentes pruebas que se realizaron como se puede ver en el video de resultados y en la seccion anterior muestra un rendimiento superior al 80% aunque es un numero muy interesante tomando en cuenta que se esta evaluando es la conformación del set de datos, tambien abre puertas hacia las mejoras que se plantearan en la siguiente sección. El entrenamiento del modelo neuronal tuvo una duración de cerca de 10 horas, con un set de datos relativamente pequeno, ya que en el grountruth y conformación de todas las imagenes para trabajo se cuentan con aproximadamente 300k imagenes etiquetadas y con sus respectivos archivos test.record y train.record, el modelo que se esta evaluando fue entrenado solo con 300 imagenes, ya que si se intentaba hacer el entrenamiento con todo el set de datos se podrian encontrar problemas y/o fallos, como primera medida lo que se desea es tener un rendimiento estable para pasar a hacer las implementaciones y mejoras del proyecto.

MEJORAS Y CONCLUSIONES

El objetivo de esta primera etapa del proyecto en el banchmarking era hacer una evaluación de los algoritmos

que se van a trabajar y las mejoras que se pueden obtener a partir de los resultados obtenidos en esta fase. Era interesante primero poder determinar si la red neuronal lograba discriminar datos reales capturados desde la camara web y evaluados con los parametros obtenidos desde el modelo entrenado con datos sinteticos, como se expreso en los resultados, aca se tienen resultados muy interesantes, ya que el algoritmo efectivamente reconoce las 3 clases con las que se entreno el modelo con una eficacia superior al noventa por ciento, pero aparecen las diferentes variaciones que se quieren hacer.

Set de datos hibridos. En este caso lo que se quiere hacer a partir de los datos creados es ampliar aun mas este numero de imagenes mediante una red GAN que se entrene con estos datos y produzca un numero mayor de imagenes, para la siguiente construcción de entrenamiento y validación se desea evaluar el comportamiento de diferentes set de datos, solo imagenes renderizadas, imagenes generadas e imagenes renderizadas e imagenes reales con imagenes renderizadas, y poder determinar de que forma en una aplicación real se pueden encontrar resultados aceptables. Ademas si esta tecnica funciona, tambien se podria implementar en los modelos de representación que se basan en la sombra de los objetos, pues una red GAN podria generar muchas imagenes nuevas, produciendo un mejor resultado.

Representación del objeto Puede que sea un problema de homografia o del modelo de representación de los objetos encontramos que en la aplicación el boundbox no correspondia en su totalidad con el objeto, para evaluar esto queremos aplicar otros conceptos de homografia y modelo de representación de objetos en el algoritmo train.py de tensorflow, tambien se puede replantear la forma en la que se esta realizando el etiquetado de las imagenes, hacer pruebas teoricas y determinar si se encuentra algun grado de error que luego es directamente proporcional con el resultado final.

Machine o Deep. Ahora es interesante plantear si se puede entrenar otro tipo de red convolucional que tenga un rendimiento igual o mayor al model actualmente entrenado, un gran inconveniente que presenta este modelo es que ante un set de datos realmente pequeno 300 elementos, demoraba cada step de entrenamiento alrededor de 1 minuto, y aunque cada entrenamiento es diferente segun su taza de perdida, se espera para tener un modelo realmente estable al menos 20.000 pasos, en este aspecto es realmente bueno considerar si el problema radica en las capacidades de la computadora, o se pueden cambiar parametros de configuración del

modelo o sencillamente es buena idea probar otro tipo de red.

REFERENCES

- [1] F. Kurtulmu and H. nal, Discriminating rapeseed varieties using computer vision and machine learning, *Expert Syst. Appl.*, vol. 42, no. 4, pp. 18801891, 2015.
- [2] P. Mcallister, H. Zheng, R. Bond, and A. Moorhead, Combining deep residual neural network features with supervised machine learning algorithms to classify diverse food image datasets, *Comput. Biol. Med.*, vol. 95, no. May 2017, pp. 217233, 2018.
- [3] S. Piri, D. Delen, and T. Liu, A synthetic informative minority oversampling (SIMO) algorithm leveraging support vector machine to enhance learning from imbalanced datasets, *Decis. Support Syst.*, vol. 106, pp. 1529, 2018.
- [4] S. Bahaadini et al., Machine learning for Gravity Spy: Glitch classification and dataset, *Inf. Sci. (Ny)*, vol. 444, pp. 172186, 2018.
- [5] Y. Song et al., Gaussian derivative models and ensemble extreme learning machine for texture image classification, *Neurocomputing*, vol. 277, pp. 5364, 2017.
- [6] W. Li, Z. Lv, D. Cosker, and Y. liang Yang, Learning system in real-time machine vision, *Neurocomputing*, vol. 288, pp. 12, 2018.
- [7] A. Ali and F. Yangyu, Unsupervised feature learning and automatic modulation classification using deep learning model, *Phys. Commun.*, vol. 25, pp. 7584, 2017.
- [8] S. Jain, E. Kinber, and F. Stephan, Automatic learning from positive data and negative counterexamples, *Inf. Comput.*, vol. 255, pp. 4567, 2017.
- [9] A. Kamarlis and F. X. Pernafeta-Bold, Deep Learning in Agriculture: A Survey, *Comput. Electron. Agric.*, vol. 147, no. 1, pp. 7090, 2018.
- [10] C. Affonso, A. L. D. Rossi, F. H. A. Vieira, and A. C. P. de L. F. de Carvalho, Deep learning for biological image classification, *Expert Syst. Appl.*, vol. 85, pp. 114122, 2017.
- [11] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, A survey on deep learning for big data, *Inf. Fusion*, vol. 42, no. August 2017, pp. 146157, 2018.
- [12] I. J. Goodfellow, J. Pouget-abadie, M. Mirza, B. Xu, and D. Warde-farley, Generative Adversarial Nets, pp. 19.
- [13] A. Odena, C. Olah, and J. Shlens, Conditional Image Synthesis with Auxiliary Classifier GANs, 2017.
- [14] M. Hohenfellner, B. Hadaschik, and J. Radtke, Adversarial Networks for the Detection of Aggressive Prostate Cancer, 2017.