# Redes neuronales adversarias para el reconocimiento de malezas

Autor: Baruffaldi Juan Manuel, baruffaldi.jm@gmail.com
Director: Lucas Uzal, uzal@cifasis-conicet.gov.ar

Facultad de Ciencias Exactas, Ingeniería y Agrimensura. Universidad Nacional de Rosario
CIFASIS, CONICET – UNR

**Resumen** Se aborda el problema de reconocimiento de malezas en video para poder realizar una fumigación selectiva de la maleza sobre campo con cultivo. El sistema de reconocimiento propuesto es compatible además con la implementación de técnicas de robótica para remover la maleza con actuadores mecánicos sin el uso de agroquímicos. El problema es abordado con técnicas de Deep Learning, donde los datos de entrenamiento son filmaciones del campo con la presencia de cultivo y maleza. El sistema de visión propuesto está basado en *Convolutional Neural Networks* (CNN). Se utilizó la técnica de *Generative Adversarial Networks* (GAN) para hacer un pre-entrenamiento no supervisado del modelo de modo de explotar la gran cantidad de imágenes que se obtienen a partir de secuencias de video. Luego se entrena en forma supervisada con una mínima cantidad de datos etiquetados para especializar el modelo. Se analizan y comparan resultados de distintos métodos utilizados y su aporte en el reconocimiento. Se combinan dos redes discriminantes de DCGAN y se utiliza una SVM en la última capa de la red entrenada sobre datos etiquetados utilizando *Data Augmentation* para lograr mayor robustez.

## 1. Introducción

En este proyecto se aborda el problema de reconocimiento de malezas en video utilizando técnicas de Deep Learning. Esto se realiza en el contexto de campo sembrado con soja y maleza en la misma altura de crecimiento, con la finalidad de distinguir la maleza del cultivo para poder eliminarla. Dentro de la rama de Deep Learning se utilizó Redes Neuronales Convolucionales (Convolutional Neural Networks), y en particular Redes Adversarias Generativas (Generative Adversarial Networks) para hacer un pre entrenamiento no supervisado del modelo. Luego se entrena en forma supervisada con una mínima cantidad de datos etiquetados para especializar el modelo en el problema de detección de malezas entre cultivo de soja.

### 1.1. Problemática

El uso constante y creciente de agroquímicos para controlar malezas en los cultivos genera un círculo vicioso: a mayores dosis de herbicidas, más resistencia se desarrolla al producto. En nuestro país se aplica tradicionalmente de manera uniforme en la totalidad del lote, independientemente de la existencia o no de malezas, por lo que su uso es cada vez más ineficiente, costoso y perjudicial para el medioambiente y la salud humana [6].
Según datos recientes de la Cámara de Sanidad Agropecuaria y Fertilizantes (CASAFE) [32], en los últimos 22 años el consumo de herbicidas aumentó un 858 %, mientras que la superficie cultivada se incrementó en un 50 % y el rendimiento de los cultivos, sólo un 30 % [6]. Esto muestra una necesidad de encontrar una solución al incremento del uso de agroquímicos con el fin de reducir costos y cuidar el medio ambiente.
Investigadores del Instituto Nacional de Tecnología Agropecuaria (INTA) de Castelar, desarrollaron un detector de malezas mediante sensores ópticos para la aplicación sitio específica de los herbicidas. Desarrollaron 2 sensores, uno que detecta la presencia de color verde y otro por reflejo de luz infrarroja. Con este último se han logrado reducciones en el uso de herbicidas de entre el 30 y el 70 % con niveles de precisión en la detección del 95 %, dependiendo esto último del tipo cultivo y de la maleza [8]. La desventaja de estos métodos es la de no poder distinguir la maleza del cultivo. Sin embargo, permiten realizar reducciones importantes en los

volúmenes de herbicidas aplicados al control de malezas cuando se realiza la práctica del barbecho químico [7].
El uso indiscriminado de herbicidas ha logrado que las malezas sean cada vez más resistentes, teniendo que aumentar la dosis para lograr eliminarlas. Un estudio realizado por docentes de la Facultad de Agronomía de la Universidad de Buenos Aires (FAUBA), estima que por año se destinan unos 1300 millones de dólares para combatir las malezas que afectan a la zona núcleo [11].
El mercado actual necesita poder realizar una fumigación inteligente, diferenciando la maleza del cultivo en las distintas etapas de crecimiento y a una velocidad de respuesta aceptable dada la velocidad de trabajo del fumigador. Sumado a esto el creciente uso de drones favorece la utilización del reconocimiento por video abriendo camino a nuevas aplicaciones de esta tecnología.Los beneficios radican en el claro ahorro del uso del agua, cuidado del medio ambiente y reducción de costos para el productor agropecuario.

### 1.2. Antecedentes

Dentro del Grupo de Aprendizaje Automático y Aplicaciones del CIFASIS existen antecedentes de la aplicación de Deep Learning al reconocimiento de especies vegetales. Se está trabajando en el diseño de un método automático para la clasificación de especies y variedades vegetales agrícolas en Argentina, analizando exclusivamente la información contenida en imágenes de las nervaduras de las hojas. Los primeros resultados se obtuvieron mediante técnicas tradicionales de Visión por Computador (Computer Vision) y Aprendizaje Automático (Machine Learning) [16,17,18]. Este problema se abordó luego con Aprendizaje Profundo (Deep Learning), mejorando los resultados previamente obtenidos [19]. Sin embargo estas técnicas no son tan efectivas para la clasificación masiva de especies vegetales en tiempo real ya que requieren una adquisición cuidadosa por medio de scanners de la imagen de la hoja.

Las soluciones existentes en el mercado, en su mayoría, radican en la detección del pigmento verde o la cantidad de luz por medios de sensores ópticos. Estos detectores (técnicamente llamados weedsekers) funcionan en tiempo real, instalado en un equipo pulverizador autopropulsado o de arrastre. Entran en acción en la instancia de barbecho químico, previa a la siembra del cultivo, cuando se controlan las malezas con agroquímicos pre-emergentes. No está indicado para utilizar con los cultivos, ya que no diferencia las malezas del cultivo en cuestión. Por su configuración modular, también puede servir para aplicaciones en los entre-surcos de plantaciones de maíz y de caña de azúcar [7,9].
Se definen a continuación las 3 técnicas más relevantes.
**Detección de color:** Una aproximación muy sencilla para lograr una pulverización selectiva radica en detectar la presencia del color verde. Con esta simple operación estaríamos en condiciones de distinguir las malezas verdes del suelo y, de esta manera, poder pulverizar solamente los sectores que poseen malezas. Este método resultaría muy beneficioso para aquellos campos que implementan la siembra directa, minimizando el volumen de herbicidas aplicado al control de malezas en el barbecho. Con el método de detección de color se consigue una clara distinción entre la maleza y el suelo. Sin embargo, por las características propias del sensor no es posible distinguir entre el cultivo y la maleza [7].
**Detección del color rojo y cercano al infrarrojo (RR/NIR):** Existen varios trabajos enfocados en la detección de malezas mediante la cantidad de luz reflejada por las mismas al ser iluminadas. Fundamentalmente se han realizado estudios para determinar la cantidad de luz reflejada cerca del infrarrojo (NIR) y en el rango del rojo (RR) [10]. Dado que el suelo y las plantas reflejan cantidades distintas de ambos espectros, es posible utilizar esta característica para realizar una clasificación [8]. En lo que al procedimiento respecta, la idea es similar. Lo único que se modifica es la forma de detectar las malezas, o sea el sensor. Las conclusiones para este método son idénticas a las del método anterior, haciendo hincapié en su simple diseño y la desventaja de no poder distinguir la maleza del cultivo [7].
**Detección mediante visión artificial:** Los sistemas de visión artificial están basados en el uso de cámaras de video o fotográficas. En este caso las imágenes pueden ser tomadas con una cámara de película convencional para su posterior procesamiento con el algoritmo de detección correspondiente. También pueden utilizarse fotografías del campo georeferenciadas antes de realizar la pulverización, para luego obtener un mapa de la

ubicación de las malezas y ser cargados en una pulverizadora que cuente con un GPS. Trabajar en tiempo real implica tomar las imágenes, procesarlas, detectar las malezas y actuar consecuentemente mientras el equipo de pulverización se encuentra en movimiento [7].

## 2. Marco teórico

El trabajo se enmarca en la rama de Machine Learning [1] y en particular nos introduciremos dentro de las redes neuronales convolucionales [25] y sus distintas formas de aprendizaje (supervisado, no supervisado y semi supervisado [20,21,22]). Utilizaremos el reciente algoritmo de redes adversarias generativas [3,5,4] por sus interesantes resultados en problemas no supervisados. Este algoritmo hace uso de redes neuronales profundas [2] que introduciremos en esta sección.

### 2.1. Redes neuronales convolucionales

Los redes neuronales convolucionales (conocidas como CNN [25]) están diseñados para procesar datos que vienen en forma de múltiples matrices, por ejemplo una imagen en color compuesta de tres matrices 2D que contienen intensidades de píxeles en los tres canales de color. Muchas modalidades de datos están en forma de múltiples matrices: 1D para señales y secuencias, incluyendo lenguaje; 2D para imágenes o espectrogramas de audio; Y 3D para video o imágenes volumétricas. Existen cuatro ideas clave detrás de CNN que aprovechan las propiedades de las señales naturales: conexiones locales, pesos compartidos, uso y agrupación de muchas capas.

La arquitectura de un CNN típico se estructura como una serie de etapas. Las primeras se componen de dos tipos de capas: capas convolucionales y capas de agrupación. Las unidades en una capa convolucional están organizadas en mapas de características, dentro de los cuales cada unidad está conectada a parches locales en los mapas de características de la capa anterior a través de un conjunto de pesos denominado banco de filtros. El resultado de esta suma ponderada local se pasa entonces a través de una no linealidad tal como una ReLU (Rectified Linear Unit) [1]. Todas las unidades de un mapa de funciones comparten el mismo filtro. Diferentes mapas de características en una capa utilizan filtros diferentes.
La razón de esta arquitectura es doble. En primer lugar, en los datos de la matriz, como en las imágenes, los valores cercanos suelen estar altamente correlacionados, formando características locales distintivas que se detectan fácilmente. En segundo lugar, las estadísticas locales de imágenes y otras señales son invariantes traslacionales. En otras palabras, si un patrón puede aparecer en una parte de la imagen, podría aparecer en cualquier lugar, de ahí la idea de unidades en diferentes lugares compartiendo los mismos pesos y detectando el mismo patrón en diferentes partes de la matriz. Matemáticamente, la operación de filtrado realizada por un mapa de características es una convolución discreta, de ahí el nombre (ver figura 1).

### 2.2. Aprendizaje supervisado, no supervisado y semi supervisado

La forma más común de aprendizaje automático, profundo o no, es el aprendizaje supervisado. Imagínese que queremos construir un sistema que pueda clasificar las imágenes según su contenido, por ejemplo una casa, un automóvil, una persona o una mascota. Primero recopilamos un gran conjunto de datos de imágenes de casas, autos, personas y mascotas, cada uno etiquetado con su categoría. Durante el entrenamiento, la máquina muestra una imagen y produce una salida en forma de un vector de núcleos, uno para cada categoría. Queremos que la categoría deseada tenga la puntuación más alta de todas las categorías, pero es poco probable que suceda antes del entrenamiento [20].

[1] $f(x) = \text{máx}(0, x)$

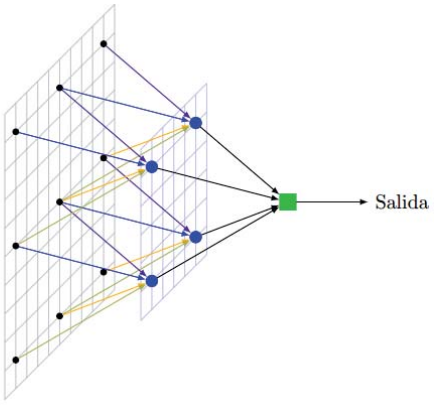


**Figura 1.** Arquitectura Redes Convolucionales

Se calcula una función objetivo que mide el error (o distancia) entre las puntuaciones de salida y el patrón de puntuaciones deseado. La máquina modifica entonces sus parámetros ajustables internos para reducir este error. Estos parámetros ajustables, a menudo llamados pesos, son números reales que pueden ser vistos como ”perillas”que definen la función de entrada-salida de la máquina. En un sistema de aprendizaje profundo típico, puede haber millones de estos pesos ajustables, y millones de ejemplos etiquetados con los cuales entrenar la máquina.
Por su parte el aprendizaje no supervisado (aprender de datos no etiquetados) tuvo un efecto catalítico al revivir el interés por el aprendizaje profundo, pero desde entonces ha sido ensombrecido por los éxitos del aprendizaje puramente supervisado. Se espera que el aprendizaje sin supervisión sea mucho más importante a largo plazo, al igual que la combinación de los mismos (es decir, aprendizaje semi-supervisado). El aprendizaje humano y animal es en gran medida sin supervisión: descubrimos la estructura del mundo observándola, no por el nombre de cada objeto [21].
El aprendizaje semi-supervisado es una clase de tareas y técnicas de aprendizaje supervisadas que también hacen uso de datos no etiquetados para el entrenamiento, típicamente una pequeña cantidad de datos etiquetados con una gran cantidad de datos no etiquetados [22].
Se ha descubierto que con el uso de datos no etiquetados junto con una pequeña cantidad de datos etiquetados se puede obtener una muy buena precisión. La recopilación de datos no etiquetados es poco costosa en relación con los datos etiquetados. A menudo los datos etiquetados son escasos y los datos no etiquetados son abundantes. En tales situaciones, se puede usar el aprendizaje semi-supervisado. Además, gran parte del aprendizaje humano implica una pequeña cantidad de instrucción directa (datos etiquetados) combinada con grandes cantidades de observación (datos no etiquetados).

### 2.3. Redes Adversarias Generativas

Los modelos generativos son uno de los enfoques más prometedores para trabajar en problemas donde no se poseen suficientes datos etiquetados. Para formar un modelo generativo primero recolectamos una gran cantidad de datos en algún dominio y luego formamos un modelo para generar datos sintéticos que resultan indistinguibles de los datos reales. Estos modelos, en el largo plazo, tienen el potencial de aprender automáticamente las características naturales de un conjunto de datos, ya sean categorías o desentramar los factores de variación en los datos [2].
Los métodos de aprendizaje profundo han presentado grandes avances en la generación de imágenes. Para esto existen distintos métodos, aunque en este trabajo utilizaremos el método GAN.

Las redes adversarias generativas (conocidas como GAN por sus siglas en ingles) [3] plantean el proceso de entrenamiento como un juego entre dos redes separadas: una red generadora y una segunda red discriminatoria que trata de clasificar las muestras como procedentes de la distribución verdadera $P(x)$ o la distribución del modelo $p'(x)$. Cada vez que el discriminador nota una diferencia entre las dos distribuciones, el generador ajusta ligeramente sus parámetros para hacerlo desaparecer, hasta que al final (en teoría) el generador reproduce exactamente la distribución de datos verdadera y el discriminador está adivinando al azar, incapaz de encontrar una diferencia.
El entrenamiento adversario es complicado porque se tiene que optimizar para un generador de imágenes y un discriminador al mismo tiempo. Este tipo de optimización es difícil, y si el entrenamiento no fuera estable, no se encontraría este punto de equilibrio entre modelos.

### 2.4. Entrenando un modelo generativo

Supongamos que se utilizó una red recién inicializada para generar 200 imágenes, donde el primer modelo generativo recibe una secuencia de números aleatorios como entrada. La pregunta es: ¿cómo debemos ajustar los parámetros de la red para producir muestras un poco más creíbles? Nótese que no estamos en un entorno supervisado simple y no tenemos ningún objetivo explícito deseado para nuestras 200 imágenes generadas; simplemente queremos que parezcan reales. Un enfoque inteligente en torno a este problema es seguir el enfoque de la red adversarial generativa (GAN). Aquí se introduce una segunda red discriminadora (normalmente una red neural convolucional estándar) que trata de clasificar si una imagen de entrada es real o generada. Por ejemplo, podríamos alimentar las 200 imágenes generadas y 200 imágenes reales en el discriminador y entrenarlo como un clasificador estándar para distinguir entre las dos fuentes. Pero además de eso -y aquí está la clave- también podemos aplicar un algoritmo de retropropagación [23] a través del discriminador y del generador para encontrar cómo debemos cambiar los parámetros del generador para hacer que sus 200 muestras sean un poco más similares a las reales para el discriminador. Estas dos redes están por lo tanto encerradas en una batalla: el discriminador está tratando de distinguir las imágenes reales de las imágenes falsas y el generador está tratando de crear imágenes que engañen al discriminador en esta tarea. Al final, la red generadora estará emitiendo imágenes que son indistinguibles de imágenes reales para el discriminador.
Las muestras del generador comienzan ruidosas y caóticas, y con el tiempo convergen para tener estadísticas de imágenes más plausibles (ver figura 2).

Estos modelos no tienen capacidad, ni están diseñados, para almacenar datos. Eventualmente, el modelo puede descubrir muchas regularidades más complejas: que hay ciertos tipos de fondos, objetos, texturas, que ocurren en ciertos arreglos probables.

Además de la finalidad de generar imágenes, se introduce un enfoque para el aprendizaje semi-supervisado con GAN que implica que el discriminador produzca una salida adicional orientada a clasificar los datos de entrada. Este enfoque permite obtener resultados de última generación en MNIST [34], SVHN [35] y CIFAR-10 [36] en entornos con muy pocos ejemplos etiquetados. En el MNIST, por ejemplo, se obtiene una precisión del 99.14 % con sólo 10 ejemplos etiquetados por clase con una red neuronal completamente conectada - un resultado que está muy cerca de los resultados más conocidos con enfoques completamente supervisados usando los 60.000 ejemplos etiquetados disponibles. Esto es muy prometedor porque la clasificación manual de ejemplos para entrenar puede ser bastante costosa en la práctica [13].

### 2.5. Redes convolucionales profundas generativas adversarias

Uno de estos modelos recientes es la red DCGAN (Deep Convolutional Generative Adversarial Networks) de Radford [4]. Este modelo introduce una clase de CNNs llamadas redes convolucionales generativas adver-

**Figura 2.** Ejemplos (DCGAN) [5]

sarias y demuestra que son un fuerte candidato para el aprendizaje sin supervisión.

Radford propone y evalúa un conjunto de restricciones sobre la topología arquitectónica Convolutional de GANs que los hacen estables para entrenar en la mayoría de los escenarios. Utiliza a los discriminadores capacitados para tareas de clasificación de imágenes, con otros algoritmos no supervisados. Visualiza los filtros aprendidos por GANs y muestra empíricamente que los filtros específicos han aprendido a dibujar objetos específicos. Por último muestra que los generadores tienen interesantes propiedades aritméticas vectoriales que permiten una fácil manipulación de muchas cualidades semánticas de las muestras generadas [4].
Esta red toma como entrada 100 números aleatorios muestreados a partir de una distribución uniforme (nos referimos a ellos como un código, o variables latentes, en rojo) y emite una imagen (un arreglo de píxeles de 64x64x3, en verde). Cuando la entrada al generador se cambia de forma incremental, las imágenes generadas también lo hacen; esto muestra que el modelo ha aprendido características para describir el dataset, en lugar de simplemente memorizar algunos ejemplos.
La red (en amarillo) está formada por componentes de red neuronal convolucional estándar, tales como capas deconvolucionales (reverso de capas convolucionales), capas totalmente conectadas, etc (ver figura 3).

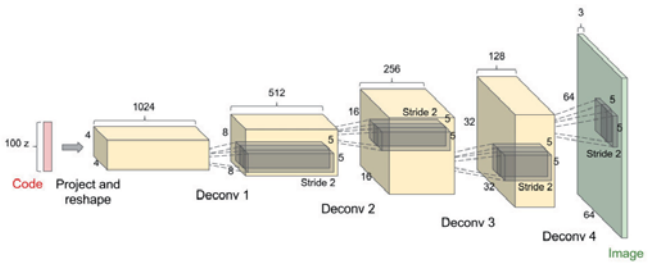


**Figura 3.** Aquitectura de Modelos Generativos (DCGAN)

DCGAN, como la mayoría de los modelos generativos, se inicializa con pesos aleatorios, por lo que un código aleatorio conectado a la red generaría una imagen completamente aleatoria. Sin embargo, la red tiene alrededor de 35 millones de parámetros que se pueden ajustar, y el objetivo es encontrar una configuración de estos parámetros que hace que las muestras generadas a partir de códigos aleatorios se parezcan a los datos de entrenamiento. O para decirlo de otra manera, se quiere que la distribución del modelo coincida con la distribución de datos verdaderos en el espacio de las imágenes.

## 3. Metodología

La construcción del dataset de entrenamiento y la elección de los métodos de aprendizaje automáticos considerados están orientados a obtener un método tal que dada una imagen tomada sobre el campo sembrado pueda detectar automáticamente las regiones de la imagen donde hay presencia de maleza. Es decir que el objetivo es poder etiquetar por regiones la imagen completa. Por este motivo trabajamos con parches de 64x64 píxeles (sobre una resolución total de 1920x1080 para la imagen completa) sobre los cuales se define el problema de clasificación (presencia o no de maleza o soja). Se propone evaluar además el aporte del aprendizaje no supervisado sobre parches sin etiquetar utilizando distintas estrategias de aprendizaje semi-supervisado. Adicionalmente se desea evaluar si la clasificación de la región de 64x64 puede ser mejorada incorporando información de contexto, es decir información de la vecindad del parche a clasificar.
La construcción del dataset y los métodos propuestos que se detallan a continuación están motivados por estas consideraciones.

### 3.1. Conjunto de datos

Al no conseguirse bases de datos para atacar esta problemática, se avanzó en la construcción de un dataset de filmaciones de campo sembrado con soja en distintas etapas de crecimiento (desde fase V6 a R2 [24]) siempre dentro de la etapa de crecimiento que nos interesa, es decir, cuando el cultivo está fuera de la tierra y antes de que cierre el surco. Para la misma se realizaron distintas grabaciones con distintos dispositivos en distintos momentos del día para obtener variaciones en la luminosidad. Las grabaciones se realizaron a distintas velocidades y en distintos lotes, con mayor o menor presencia de maleza y con y sin presencia de rastrojo. Generar la base de datos de entrenamiento implicó ir al terreno a capturar con videos los posibles escenarios a reconocer y desarrollar una herramienta que simplifique el etiquetado de malezas sobre las imágenes adquiridas. Aún así la cantidad de datos etiquetados, dada las características del problema, no son suficientes para entrenar el modelo profundo en forma completamente supervisada. Por esta razón se decidió utilizar un esquema de aprendizaje semi-supervisado.

Se trabajó con videos de 1920x1080 a 30fps. Para cada frame se realiza el reconocimiento seleccionando una región de interés de 576 píxeles de alto y 1472 de ancho centrado en la imágen; es decir, la región de interés de cada frame tiene un tamaño de 1472x576 (ver figura 4). Se determinó de esta forma ya que el foco de la cámara mantiene muy buena nitidez en esta región y desenfoca hacia el horizonte. En total, en cada región de interés hay 207 parches de 64x64 a analizar (23x9).

Para la etapa supervisada, se consideraron los parches de 64x64, dentro del área de interés, de cada frame, con sus respectivos entornos. Se tomaron frames aleatorios no consecutivos para lograr una buena variedad. Para los distintos algoritmos se realizó validación cruzada [26] para seleccionar los mejores parámetros. Hay que destacar que los videos utilizados para la etapa de test no fueron utilizados en ninguna etapa de entrenamiento.
La forma de etiquetación fue manual por medio de una plataforma web de etiquetado colectivo creada exclusivamente para este proyecto (ver figura 5). Las personas que realizaron el etiquetado están relacionadas de alguna forma con el sector agropecuario, como por ejemplo productores, pero no necesariamente ingenieros

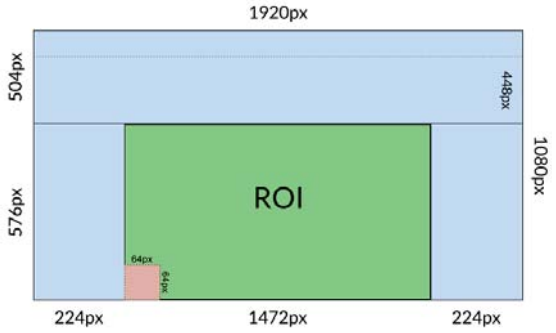


**Figura 4.** Región de interés sobre cada frame

agrónomos. Participaron en total 23 personas.
En esta web de etiquetado se visualizaron frames al azar donde se pedía etiquetar de forma aleatoria maleza, soja o entre-surco; para generar un etiquetado aleatorio del dataset pero mostrando el frame completo con el fin de que la persona tenga información de contexto para tomar la decisión. El tamaño de cada sección a seleccionar es de 64x64 y coincide con los parches de la red DCGAN. De forma automática los entornos correspondientes a cada parche son etiquetados con la misma etiqueta que el parche.


**Figura 5.** Web para etiquetado colaborativo de parches en frames aleatorios.

Se etiquetaron 200 frames, generando 5102 parches de 64x64 etiquetados de entrenamiento y 4075 de testeo. Esto representa el 5 % de los datos no supervisados que es donde se encuentra el fuerte del reconocimiento. Se espera que esta cantidad de datos sea suficiente ya que la etapa supervisada se utiliza para refinar sólo la última capa de la red, mientras que el aprendizaje intensivo se da en la etapa no supervisada.
Para la etapa no supervisada se seleccionaron, por cada frame, 25 parches de 64x64 pixeles de ubicaciones aleatorias dentro de la región de interés. Sumado a esto para cada frame se seleccionaron, de forma aleatoria, 5 recortes de 512x512 pixeles para generar datos no supervisados asociados a los entornos de la región a clasificar (ver figura 6), que son escalados a 64x64 pixeles.

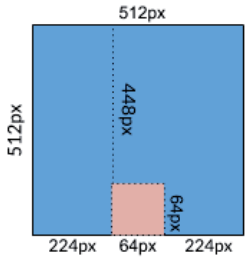


**Figura 6.** Entornos de trabajo de 512x512px

### 3.2. Métodos

Se utilizaron redes neuronales convolucionales con aprendizaje semi-supervisado utilizando GAN en la etapa no supervisada. En una primer etapa se hicieron pruebas con algoritmos k-means y PCA para tener una primer aproximación del problema.
En segunda etapa, se tomó hasta la penúltima capa de la red discriminante de aplicar DCGAN a los parches de la imagen y de igual forma al aplicar DCGAN a los entornos. Estas dos redes discriminantes, sin sus últimas capas, se combinaron en una por medio de una concatenación de matrices. Sobre esta representación se entrena una *Support Vector Machine* (SVM) [30] utilizando datos etiquetados. De esta forma se entrenaron dos redes DCGAN en 50000 iteraciones (22 épocas), una con los parches y otra con los entornos correspondientes (ver figura 7).

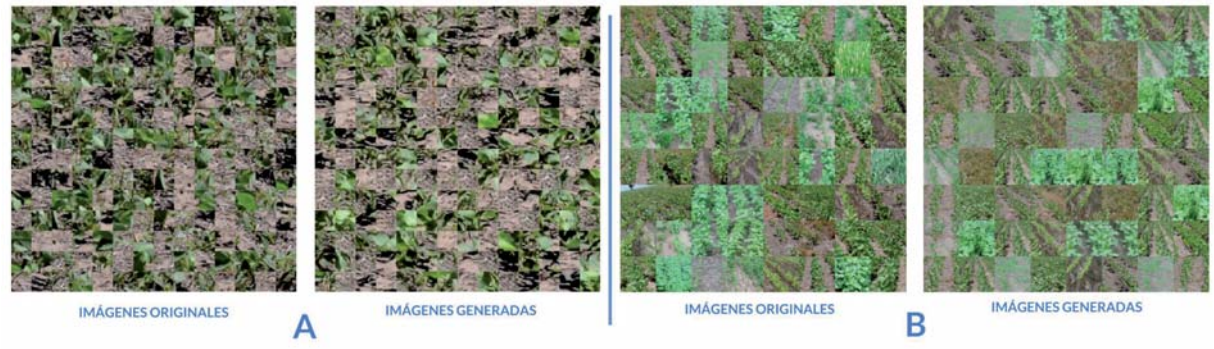


**Figura 7.** Salida de DCGAN para parches de 64x64 (A) y para entornos de trabajo (B)

Luego de esto se decidió experimentar, de forma preliminar, con un aprendizaje semi supervisado de DCGAN. En este caso al discriminante se le agrega una capa de salida extra que realiza la clasificación. El clasificador así obtenido comparte con el discriminante todas excepto la última capa. A cada iteración de DCGAN se le intercala una iteración de entrenamiento del clasificador sobre los datos etiquetados. De esta forma se entrena en forma conjunta DCGAN y el clasificador, evitando el sobreajuste de este último sobre la pequeña fracción de datos etiquetados. Utilizando una fracción de datos de validación se puede determinar en qué iteración se obtiene el mejor modelo y no necesariamente retener el correspondiente a la última iteración como en los casos anteriores. Nos referiremos a este método como DCGANCLASS.

### 3.3. Aumentación de datos

Para reforzar el aprendizaje supervisado, dado que adquirir datos etiquetados es muy costoso, se decidió utilizar la técnica de textitdata-augmentation (DA) [27,28] para mejorar el tamaño del dataset. Esta técnica se obtuvo del módulo de la librería Keras [29] que realiza pequeñas perturbaciones aleatorias sobre las imágenes basadas en rotaciones, cambios de coloración, espejados, entre otras transformaciones (ver figura 8). Se sabe que esta estrategia produce en general una mejora del desempeño en clasificación de imágenes [28].


**Figura 8.** Ejemplo de data augmentation sobre los datos

## 4. Experimentos y Resultados

Se trabajó en un cluster del CIFASIS con nodos de 8 núcleos Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz, 16gb de RAM, GeForce GTX 970, 4gb de memoria. Un entrenamiento completo de DCGAN en estas arquitecturas ronda las 24 horas. Se dispuso además de una máquina virtual de Microsoft Azure [31] de 6 núcleos Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.60GHz, 56gb de RAM, Tesla K80, 12gb de memoria.

Se utilizó Python con Lasagne [37] basado en Theano [38] para la implementación de los modelos profundos y ScikitLearn [39] para la implementación de SVM.

### 4.1. Resultados

Para evaluar lo trabajado hasta el momento y el potencial de las redes DCGAN realizamos distintos experimentos comparando cuánto aporta cada técnica sobre el problema, basados en el porcentaje de aciertos sobre el conjunto de testeo etiquetado previamente seleccionado.

A modo de motivación se decidió explorar el problema en primera instancia con el algoritmo k-means para evaluar la representación obtenida con DCGAN. Esto se realizó utilizando únicamente los datos no etiquetados, es decir que esta primer aproximación está basada únicamente en aprendizaje no supervisado.

En particular con el algoritmo k-means [39] buscando 3 clusters se espera que estos tres clusters capturen las tres categorías más evidentes en las imágenes: soja, maleza y zurco. Para esto se tomó la penúltima capa de DCGAN, aplicada sobre los parches de 64x64 sacados de los frames, y se la utilizó como entrada del algoritmo k-means buscando los 3 clusters. Esto se contrastó con el resultado de aplicar k-means por un lado, sobre la representación de píxeles crudos directamente y por otro lado sobre el resultado de aplicar PCA a los parches, para obtener la misma dimensionalidad que la penúltima capa de DCGAN [2].

[2] La dimensionalidad de DCGAN en su penúltima capa es de 1024 características y se utilizaron 2 para graficar.

Las Figuras A, B y C en 9 muestran los resultados obtenidos con k-means ($k = 3$) aplicado sobre los datos crudos, sobre PCA y sobre la representación obtenida con DCGAN, respectivamente.

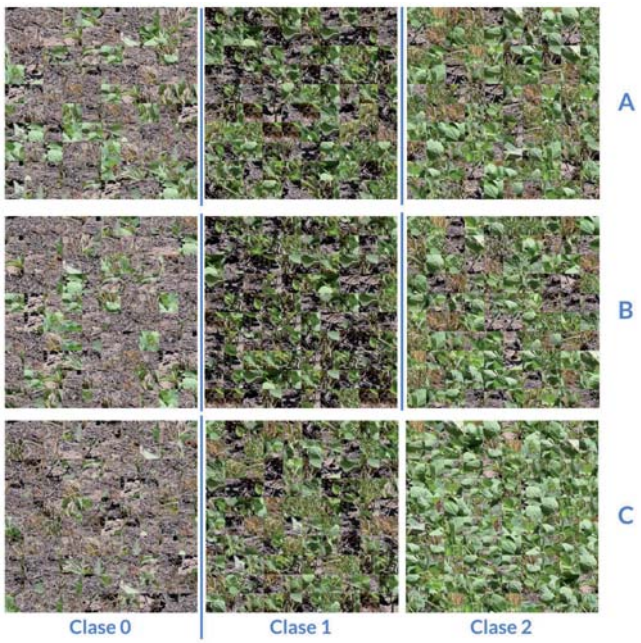


**Figura 9.** Kmeans aplicado sobre datos crudos (A), con reducción de dimensionalidad (B) y con DCGAN (C)

Se analizaron varias ejecuciones distintas y se detectó que la formación de clusters, en cantidad de elementos, varía entre un 15 % y un 32 %; comparada con la clasificación de DCGAN que se mantiene estable en todas las ejecuciones. Esto es una motivación que nos da indicio de que es una buena herramienta para seguir avanzando dado el potencial en la etapa no supervisada.

Ahora nos interesa comparar cuál es el aporte de cada una de las herramientas mencionadas anteriormente: DCGAN sobre los parches, DCGAN sobre los entornos, DCGAN + SVM con data augmentation (DA). Para este primer trabajo se decidió reducir el espacio de clases de 3 a 2, ya que lo interesante es diferenciar la maleza de todo lo demás; quedando a futuro analizar las herramientas con 3 o más clases.
Los experimentos realizados sobre los conjuntos de entrenamientos y testeos ya mencionados son:

- Aplicar SVM sobre los datos etiquetados crudos.
- Aplicar SVM + DA sobre los datos etiquetados crudos.
- Utilizar la red discriminante de DCGAN entrenada sobre parches no etiquetados y aplicarla sobre parches etiquetados aumentados con DA. Sobre la representación de la penúltima capa del discriminante entrenar una SVM.
- Concatenar las representaciones de las redes discriminantes de DCGAN entrenadas sobre entornos y sobre parches y sobre este espacio entrenar una SVM con los datos etiquetados.
- Agregar al ítem anterior DA sobre los datos etiquetados.

En la Tabla 1 se presentan los resultados basados en el porcentaje de acierto sobre el mismo conjunto de testeo (sin aplicar data augmentation sobre este conjunto).

**Cuadro 1.** Resultados de evaluación

| Accuracy | Train | Test |
|---|---|---|
| KMEANS | 55 % | 54 % |
| SVM | 99 % | 67 % |
| SVM + DA (Data Augmentation) | 100 % | 54 % |
| DCGAN parches + SVM + DA | 88 % | 83 % |
| DCGAN parches + DCGAN entornos + SVM | 93 % | 87 % |
| DCGAN parches + DCGAN entornos + SVM + DA | 91 % | 86 % |

### 4.2. DCGANCLASS

Se siguió trabajando de forma preliminar en el entrenamiento del métodos DCGANCLASS (ver Sec. 3.2). Dados los tiempos solo fue posible aplicar este aprendizaje sobre los parches, quedando a futuro aplicar está técnica sobre los entornos y sobre ambos al mismo tiempo.
El mejor modelo fue obtenido a las 28000 iteraciones obteniendo 96 % de acierto sobre el conjunto de entrenamiento, 88 % sobre el conjunto de validación y 87 % sobre test. Luego se decidió experimentar nuevamente utilizando este modelo, para contrastar los resultados antes obtenidos, sobre los siguientes casos:

- Utilizar la red discriminante de DCGANCLASS y en la última capa aplicar SVM junto con data augmentation de los datos.
- Combinar la red discriminante de DCGANCLASS sobre parches con el discriminante obtenido con DCGAN sobre entornos. Entrenar SVM sobre las representaciones concatenadas.
- Ídem anterior, con Data Augmentation.

En la Tabla 2 se muestran los resultados basados en el porcentaje de acierto siguiendo el procedimiento para los resultados anteriores.

**Cuadro 2.** Resultados Preliminares

| Accuracy | Train | Test |
|---|---|---|
| DCGANCLASS + SVM + DA | 94 % | 91 % |
| DCGANCLASS + DCGAN entornos + SVM | 98 % | 84 % |
| DCGANCLASS + DCGAN entornos + SVM + DA | 97 % | 84 % |

## 5. Conclusión

Los resultados obtenidos muestran que es posible alcanzar altos niveles de detección de malezas en imágenes de cultivos de soja sin requerir de un proceso intensivo de etiquetado manual de datos para entrenamiento. Esto está fundamentalmente basado en el entrenamiento no supervisado de redes neuronales, que permite alcanzar un 91 % de aciertos utilizando sólo un 5 % de datos etiquetados.
Concretamente, se observó que la técnica de entrenamiento no supervisado considerada basada en Redes Neuronales Adversarias captura correctamente la distribución de los datos y permite entrenar una representación de los mismos que simplifica el proceso de clasificación.
Dado que los métodos considerados son de propósito general y resultan agnósticos de la aplicación concreta,

resulta interesante de plantearse atacar otros problemas similares donde es muy costosa la adquisición de datos etiquetados, utilizando estas técnicas.
Las pruebas con la técnica de Data Augmentation no arrojaron mejoras en la clasificación con SVM. Esto se debe a que SVM tiene una complejidad cúbica respecto la cantidad de datos y limita severamente la cantidad de datos que se pueden generar con esta técnica.
En cuanto a la incorporación de información de contexto, es decir utilizar conjuntamente un campo visual más amplio para obtener la clasificación local, observamos que esta información puede mejorar el desempeño. Se requieren más experimentos para poder caracterizar más precisamente su incidencia.
Analizando los últimos resultados preliminares con DCGANCLASS se puede observar una clara mejora al utilizar otro clasificador en vez de una SVM. La posibilidad de acoplar el entrenamiento de un clasificador al entrenamiento con DCGAN es muy prometedor. Se obtuvieron mejores resultados que los alcanzados anteriormente, lo cual da un indicio de que es el siguiente camino a seguir.
Si bien los entornos empeoraron el resultado en este caso, se estima que se debe a que la red no se entrenó en simultáneo con la red de parches.

La finalidad del trabajo fue explorar las distintas alternativas para atacar el problema y ver cuál es el aporte de cada técnica. Aún así hay cosas para seguir trabajando y mejorar.

## 6. Trabajo futuro

El trabajo, enmarcado en una tesis de grado, concluye con la finalización de una red neuronal semi supervisada que incluye, aparte de todo lo analizado, la posibilidad de realizar el aprendizaje supervisado mientras se entrena DCGAN, que en este trabajo se incorpora una versión preliminar del mismo.
Sumado a esto a futuro vamos a hacer análisis sobre distintos videos de distintos campos, logrando mayor variabilidad en los datos.
A futuro se puede combinar esta técnica aplicando algoritmos de decisión sobre vecinos para determinar la presencia de maleza y técnicas de seguimiento de flujo óptico en video para una vez detectada la maleza seguirla en los siguientes frames del video.
Desde el punto de vista de la problemática se genera una solución muy prometedora a la hora de fumigar. Pero es el primer paso hacia una fumigación inteligente, o mejor dicho, hacia un control de malezas inteligente, que permita combatirlas sin aplicar producto (por ejemplo con actuadores robóticos).
Se espera que al incorporar la visión de cámaras se creen redes neuronales profundas donde no solo se pueda diferenciar la maleza del cultivo sino también diferenciar variedades de malezas entre sí para poder aplicar distintas técnicas sobre la misma.

## Referencias

1. Machine Learning, Tom Mitchell, McGraw Hill, 1997
2. Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville. MIT Press 2016. `http://www.deeplearningbook.org`
3. Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, et al. *Generative adversarial nets.* NIPS, 2014.
4. Alec Radford, Luke Metz, and Soumith Chintala. *Unsupervised representation learning with deep convolutional generative adversarial networks.* arXiv, 2015. `https://github.com/Newmu/dcgan_code`
5. Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, *Improved Techniques for Training GANs.* Goodfellow, Jun 2016.
6. Andrés Moltoni, Gerardo Masiá, *Detector de malezas para la aplicación sitio específica de herbicidas*, Innovar 2011. Laboratorio de Electrónica de INTA Castelar, Nov 2011.
7. Moltoni, A. F., Moltoni, L. A. (2006). Pulverización selectiva de herbicidas. implicancias tecnológicas y económicas de su implementación en la Argentina. Curso Internacional de Agricultura de Precisión. 6. Expo de Máquinas Precisas. 1. 2006 07 25-27, 25 al 27 de julio de 2006. Manfredi, Córdoba. AR.

8. Biller R. H. 1998. Reduced input of herbicides by use of optoelectronic sensors. Journal of Agricultural and Engineering Research 71 4 , 357-362.
9. Site-specific weed control: desempeño de un detector de maleza diseñado y construido en el instituto de ingeniería rural de inta castelar. Andrés Moltoni, Luciana Moltoni, Leonardo Venturelli, Adriana Fuica, Gerardo Masiá. Xxxv Congresso Brasileiro De Engenharia Agrícola, Agosto 2006
10. Shropshire, G. J., K. Von Bargen, and D. A. Mortensen. 1990. Optical reflectance sensor for detecting plants. In Proceedings os SPIE (the International Society for Optical Engineering): Optics in Agriculture, 7-8 November, Boston, Mass., Vol 1379:222-235. Bellinham, Wash.:SPIE Press.
11. "Malezas: se pierden USD 1300 millones", LA NACION, Sábado 07 de marzo de 2015. http://www.lanacion.com.ar/1773794-malezas-se-pierden-us-1300-millones
12. "Desarrollaron un sensor para detectar malezas en los cultivos", LA NACION, Viernes 22 de mayo de 2009, http://www.lanacion.com.ar/1130830-desarrollaron-un-sensor-para-detectar-malezas-en-los-cultivos
13. Andrej Karpathy, Pieter Abbeel, Greg Brockman, Peter Chen, Vicki Cheung, Rocky Duan, Ian Goodfellow, Durk Kingma, Jonathan Ho, Rein Houthooft, Tim Salimans, John Schulman, Ilya Sutskever, and Wojciech Zaremba. *Generative Models*. Jun 2016.
14. Yann LeCun, Yoshua Bengio and Geoffrey Hinton *Review Deep learning*. Nature, 2015.
15. Rasmus, Antti, et al. *Rasmus, Antti, et al.*. Advances in Neural Information Processing Systems. 2015.
16. Mónica G. Larese, Rafael Namías, Roque M. Craviotto, Miriam R Arango, Carina Gallo, and Pablo M Granitto. Automatic classification of legumes using leaf vein image features. Pattern Recognition, 47(1):158–168, 2014.
17. Larese, Mónica G., Bayá A. et al. Multiscale recognition of legume varieties based on leaf venation images. Expert Systems with Applications 41.10 (2014): 4638-4647.
18. Larese, Mónica G., and Pablo M. Granitto. Finding local leaf vein patterns for legume characterization and classification. Machine Vision and Applications (2015): 1-12.
19. Grinblat, G.L., Uzal, L.C., Larese, M.G., Granitto, P.M., Deep learning for plant identification using vein morphological patterns, Computers and Electronics in Agriculture 127 (2016): 418-424.
20. Kotsiantis, S. B., Zaharakis, I., Pintelas, P. (2007). Supervised machine learning: A review of classification techniques.
21. Coates, A., Ng, A., Lee, H. (2011, June). An analysis of single-layer networks in unsupervised feature learning. In Proceedings of the fourteenth international conference on artificial intelligence and statistics (pp. 215-223).
22. Zhu, X., Goldberg, A. B. 2009. Introduction to semi-supervised learning. Synthesis lectures on artificial intelligence and machine learning, 3 1, 1 130.
23. Isasi Viñuela, P. E. D. R. O., Galván León, I. M. (2004). Redes de neuronas artificiales. Un Enfoque Práctico, Editorial Pearson Educación SA Madrid España.
24. Andrade, F. H., Aguirrezábal, L. A. N., Rizzalli, R. H. (2000). Crecimiento y rendimiento comparados. Bases para el manejo del maíz, el girasol y la soja. Buenos Aires: Editorial Médica Panamericana, 61-96.
25. LeCun, Yann. LeNet-5, convolutional neural networks. Retrieved 16 November 2013.
26. Refaeilzadeh, P., Tang, L., Liu, H. (2009). Cross-validation. In Encyclopedia of database systems (pp. 532-538). Springer US.
27. https://keras.io/preprocessing/image/
28. CS231n: Convolutional Neural Networks for Visual Recognition Spring 2017, Lecture 7. http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture7.pdf
29. https://keras.io/
30. http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
31. https://azure.microsoft.com/es-es/
32. http://www.casafe.org/
33. http://inta.gob.ar/
34. http://yann.lecun.com/exdb/mnist/
35. http://ufldl.stanford.edu/housenumbers/
36. http://www.cs.toronto.edu/~kriz/cifar.html
37. https://github.com/Lasagne/Lasagne
38. http://www.deeplearning.net/software/theano/
39. http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

## A. Apendice

Ejemplos de datos de entrada de las redes. La Fig. 10 muestra un ejemplo de parche de trabajo de 64x64 pixeles. La Fig. 11 muestra un ejemplo de frame de uno de los videos utilizados.

**Figura 10.** Ejemplo de parches de trabajo de 64x64px

**Figura 11.** Ejemplo de frame