Detection and correction of deformations in UAVs' images using deep learning

Leandro Henrique Furtado Pinto Silva

Advisor: Prof. Dr. André Ricardo Backes

Co-Advisor: Prof. Dr. Mauricio Cunha Escarpinati

Programa de Pós-Graduação em Ciência da Computação (PPGCO) - FACOM/UFU

May 5, 2021

Outline

Introduction

Introduction

- Thomas Robert Malthus, a British economist, was pessimistic in his work "Essay on the Principle of Population" (1798).
- The population grew according to a geometric progression.
- The Food production grew according to an arithmetic progression.

Introduction

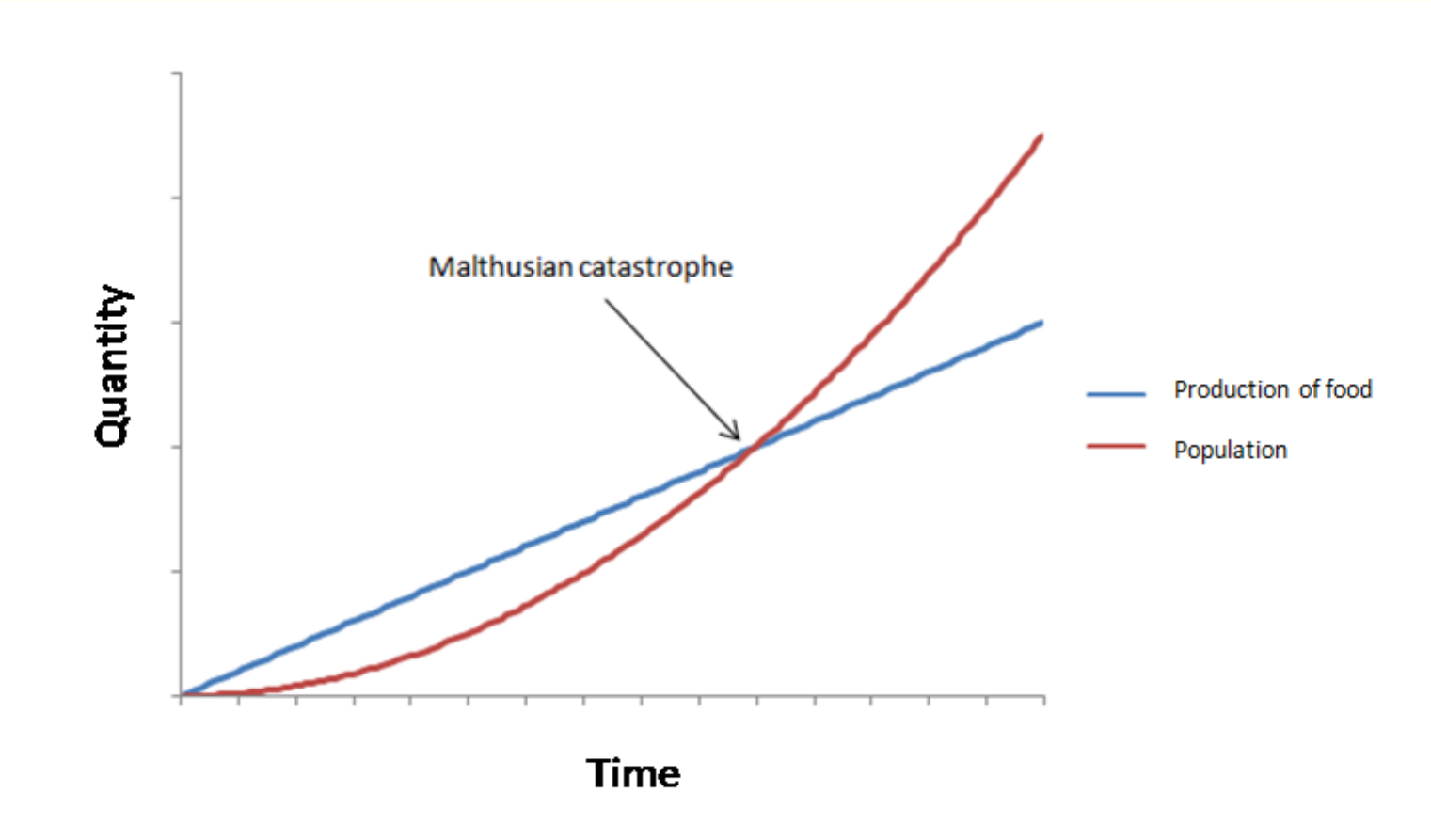


Figure: Population growth vs. food production: projections pointed to population growth in geometric progression, while food production capacity was growing in arithmetic progression.

Introduction

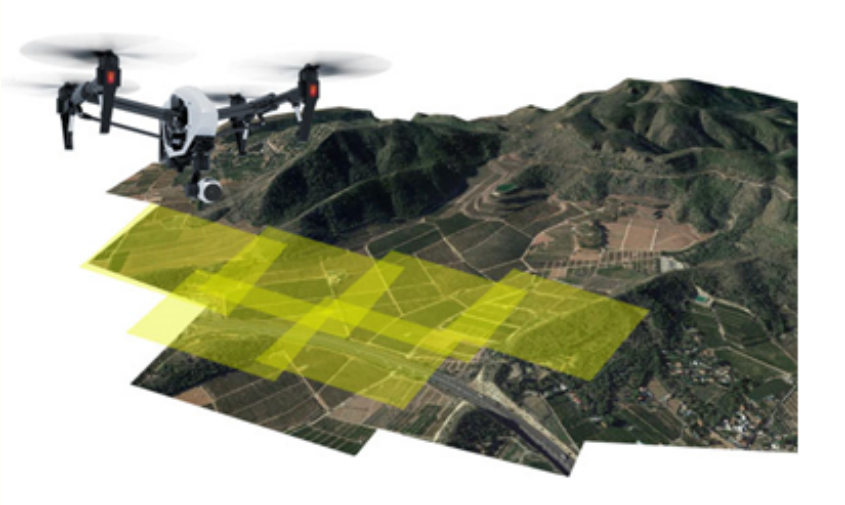
- The transformation process in agriculture on a global scale began in the middle of the 20th century.
- Development and incorporation of new production technologies.
- This process became known as the “Green Revolution”.

Introduction

- One of the main components of the current phase of agricultural research is Precision Agriculture (PA).
- PA proved to be strongly dependent on imaging and mapping technologies to estimate important agronomic characteristics.
- Given this scenario, some mechanisms for obtaining these images have stood out, in particular, Unmanned Aerial Vehicle (UAV).

Introduction

- UAVs capture images at low and medium altitudes (50 to 400m), providing a more detailed view of the region.
- The development of new sensors has enabled the acquisition and use of multispectral images in PA.



Introduction

- A multi or hyperspectral image, to be useful in agricultural applications, needs, first, to have its bands aligned.
- Co-registration presents a series of difficulties to be performed.
- Each element is represented differently in each spectrum.

Introduction

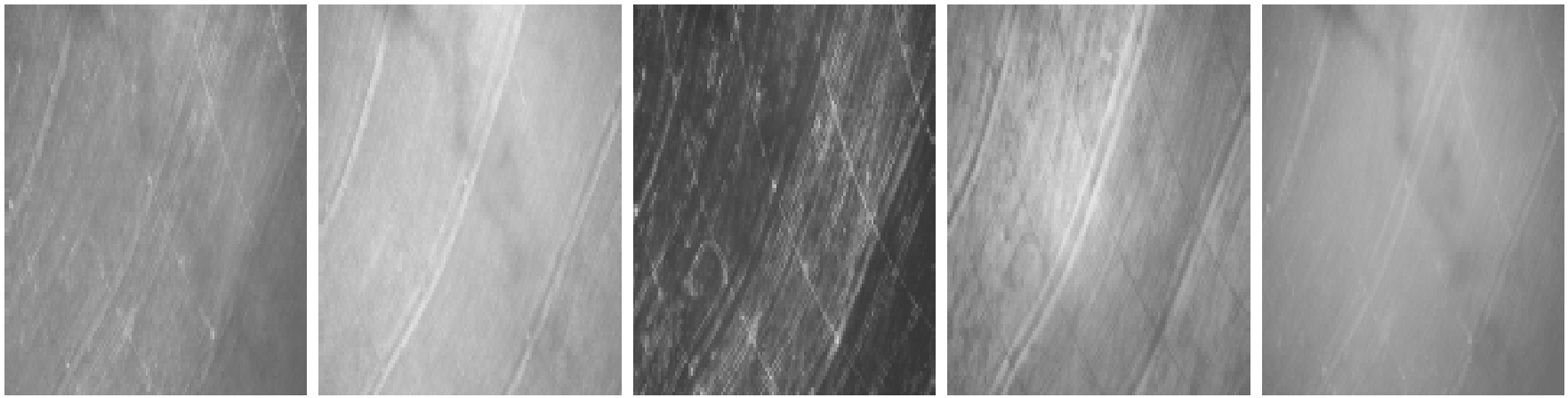


Figure: Example of an image containing all bands. Blue, Green, Red, Near-Infrared, and Red Edge, respectively.

Problem Definition

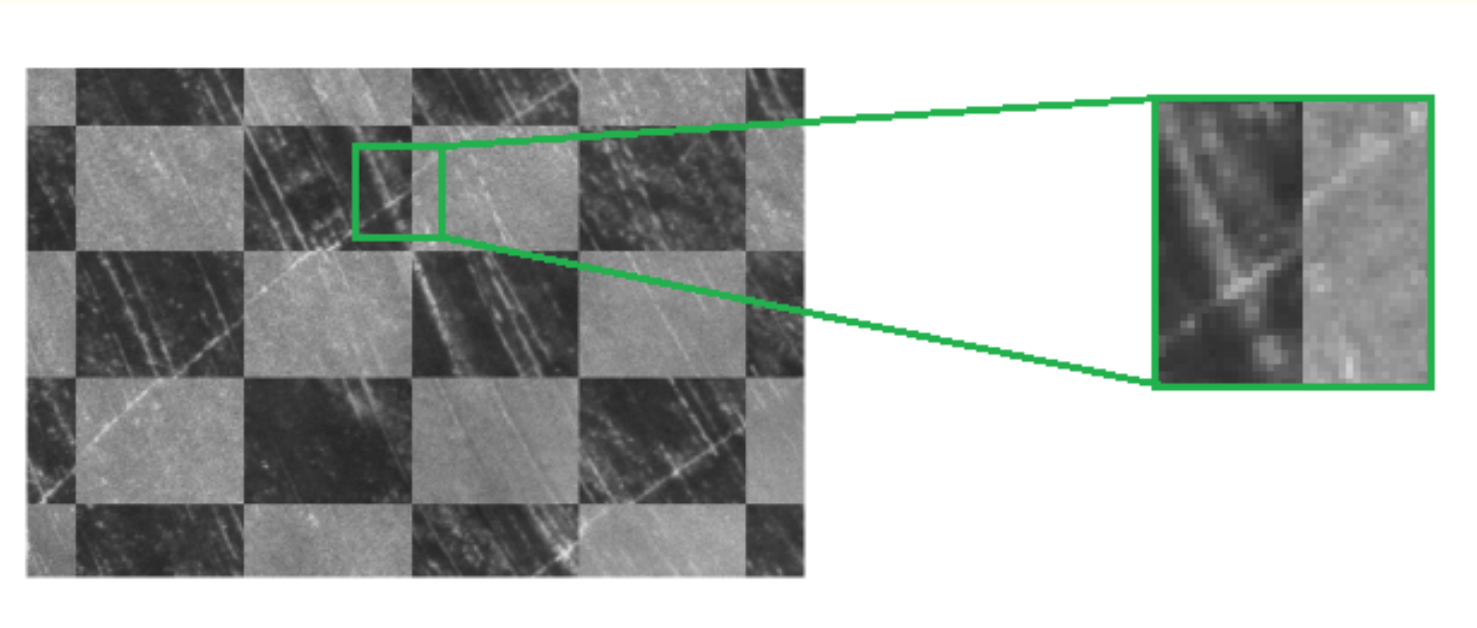


Figure: Checkerboard of two bands of the same image. Note that in highlight there is the misalignment between the bands, where the same line, in its different bands, should follow the same diffraction.

Problem Definition

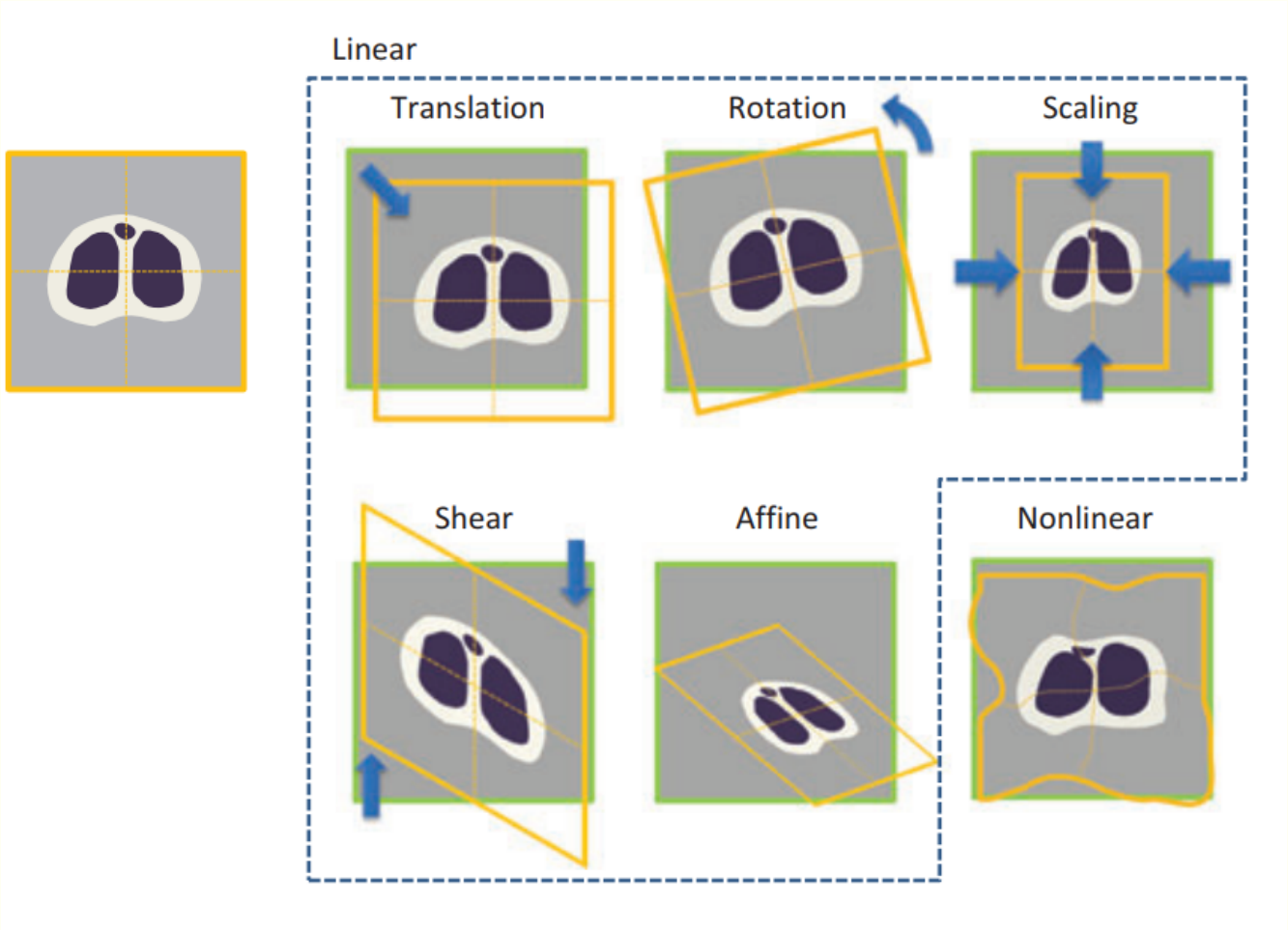


Figure: Example of linear and non-linear transformations. (Uchida, 2013).

Objectives

The general objective of this work is to investigate the ability of convolutional neural networks to identify and correct linear and non-linear deformations in images obtained by UAVs in PA.

Fundamentals

Precision Agriculture

- Precision Agriculture (PA) is considered to be the third revolution in modern agriculture.
- PA aims to divides the fields into management areas that are treated individually.
- This results in the optimization of resources, generating profit for farmers since the inputs are applied precisely, avoiding waste

UAVs in Precision Agriculture

- Currently, UAV have become popular due to their more affordable cost and the provision of high-resolution.
- One of the main characteristics of UAV is the possibility of coupling several sensors and thus making specific measurements.

UAVs in Precision Agriculture

(a)

(b)

Figure: Two types of UAV: in (a) we have a fixed-wings; in (b) we have a multi-rotor.

UAVs in Precision Agriculture

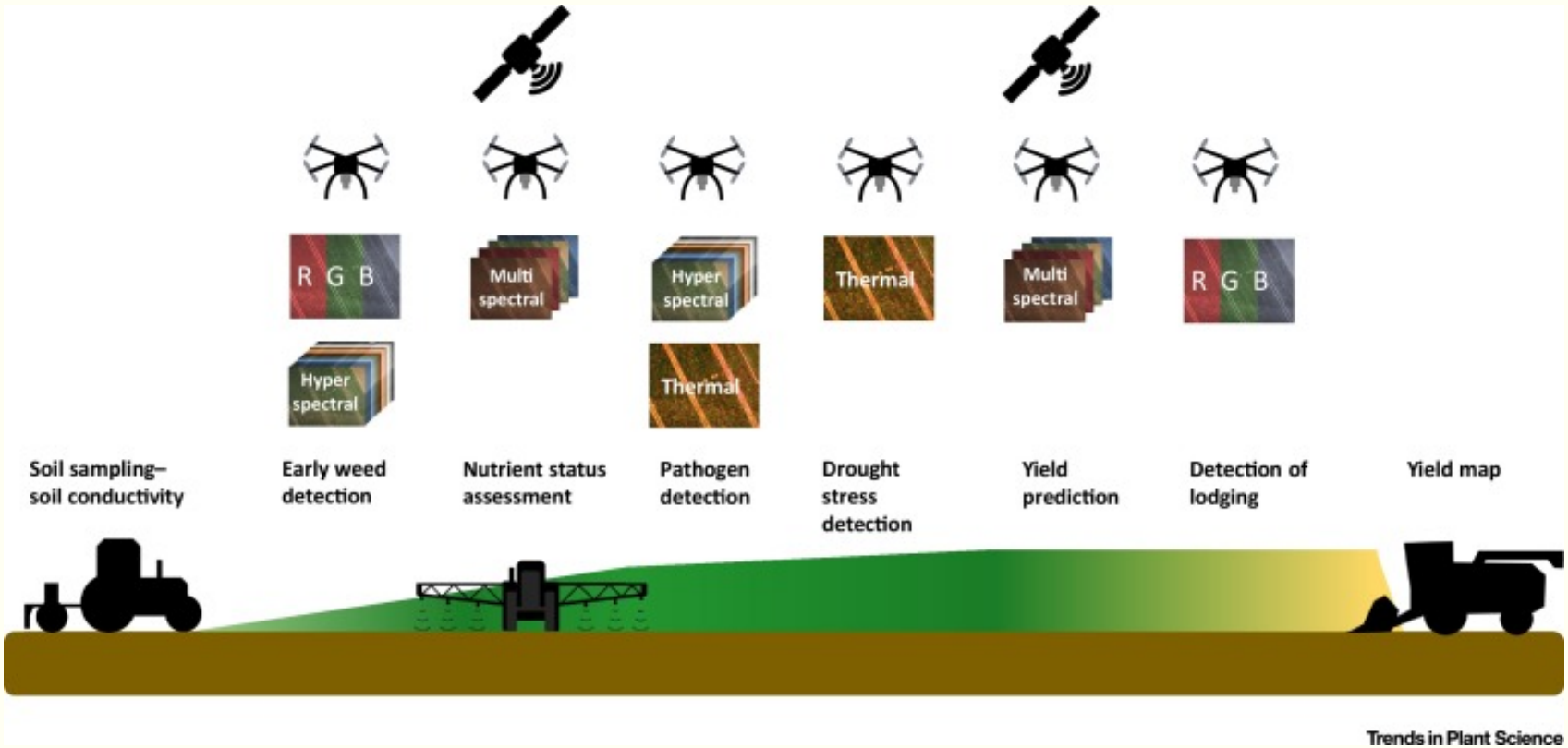


Figure: The role of UAV for assessment of farms. (MAES; STEPPE, 2019).

Neural Networks

- Neural Netwoks use a mathematical model to simulating the basic functioning of the brain.
- The neurons are activated through the composition of two functions: activation and propagation.
- The neural models can be based on:
  - **The training strategy:** supervised or unsupervied.
  - **The form of training:** incremental or in batch.
  - **The form of operation:** unidirectional or recurring.

Convolutional Neural Networks

- CNN are a category of deep learning algorithms inspired by the human learning process.
- It can explore the spatial correlations among pixels in an image to extract relevant attributes for different tasks.
- Most CNN models available in the literature are defined in terms of three types of layers.
  - Convolutional.
  - Pooling.
  - Dense.

Convolutional Neural Networks

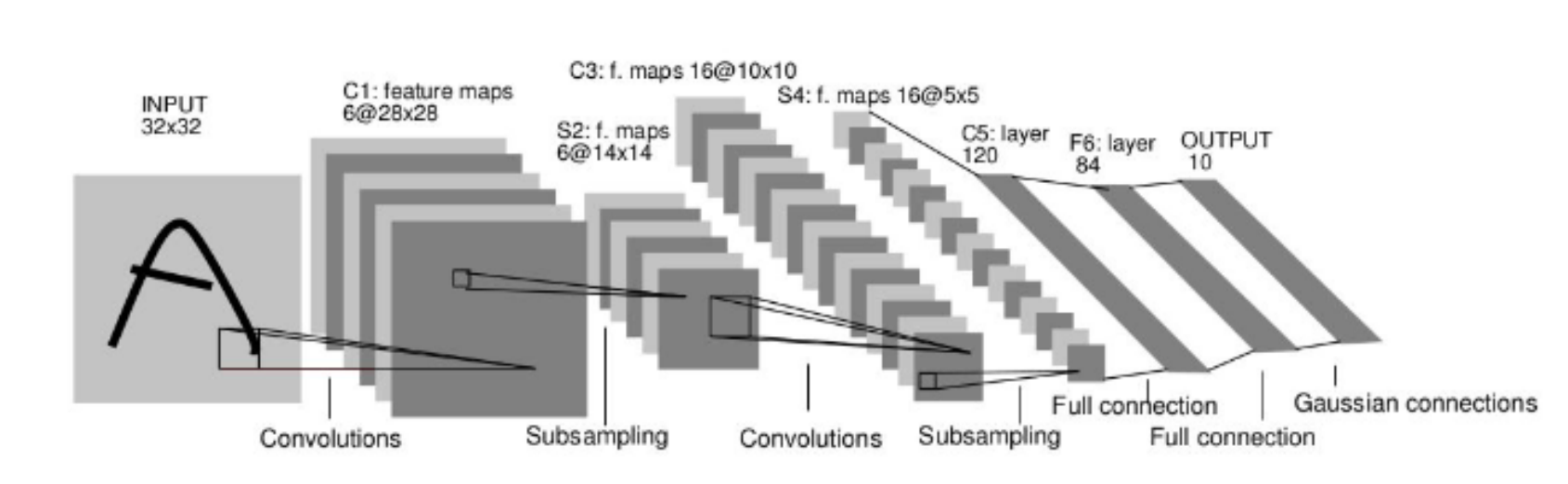


Figure: The structure of a CNN proposed by LeCun *et al.* (1990).

U-Net

- U-Net is an architecture inspired by the Fully Convolutional Network, proposed to address segmentation problems in medical image.

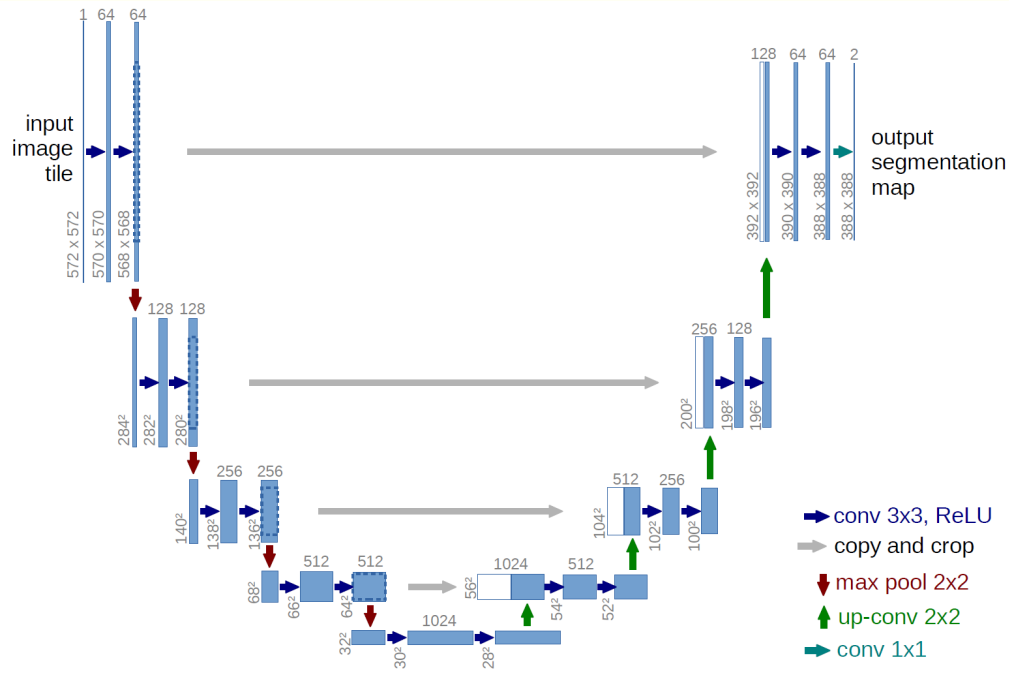


Figure: U-Net architecture (RONNEBERGER; FISCHER; BROX, 2015).

Image Registration

- The image registration process consists of a geometric transformation, which relates the coordinates of a reference image A to the coordinates of an image to be registered B.
- The vast majority of methods for image registration are composed of four basic steps:
  1. Feature Detection.
  2. Feature Matching.
  3. Construction of Mapping Function.
  4. Image Transformation.

Image Registration

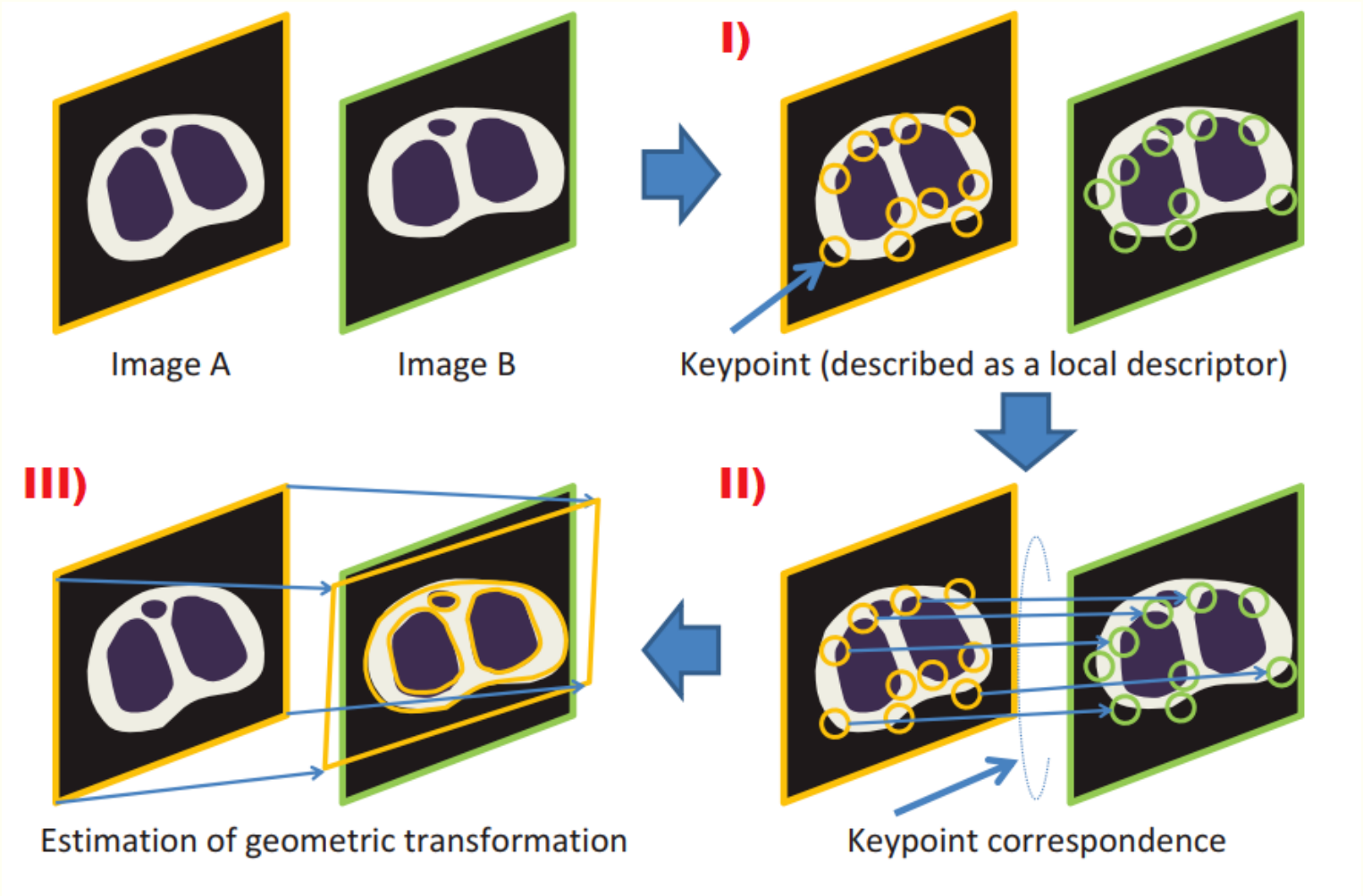


Figure: Necessary steps for registering images using keypoints. (UCHIDA, 2013)

Image Registration

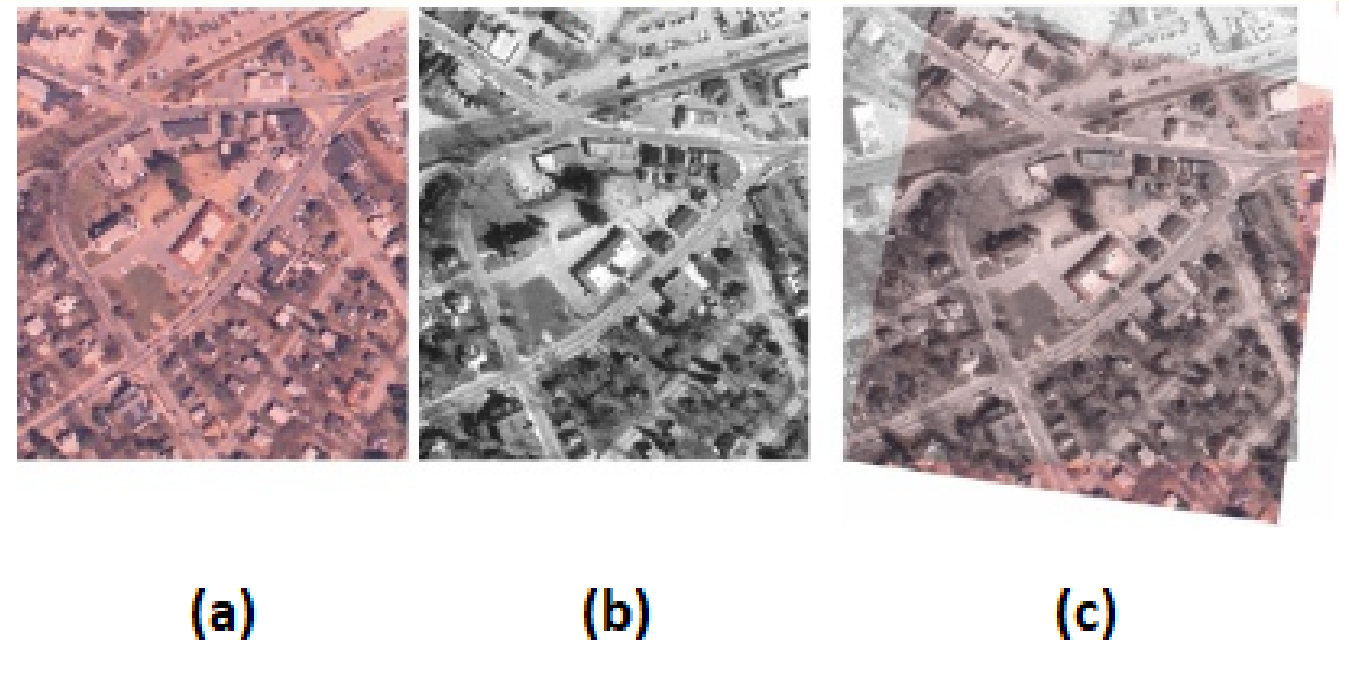


Figure: Example of transforming an image (a) to the target image (b). The result of the overlap, after transformation, is shown in (c). (MATHWORKS, 2018).

Deformation Recognition in UAVs Images Using Deep Learning

Image Dataset

- We considered two mosaics (18543×2635 and 8449×11180 pixels size) of UAVs' images.
- For each dataset, we selected grayscale patches of 128×128 pixels size.
- We discard patches no significant visual information, i.e., number of pixels (n) with value equals to 0.
  - If $n < 10$, the patch is considered for the composition of the dataset; otherwise, the patch is discarded.
- Therefore, we built two datasets, DS1 and DS2, which have, respectively, 3353 and 2365 images.

Image Dataset

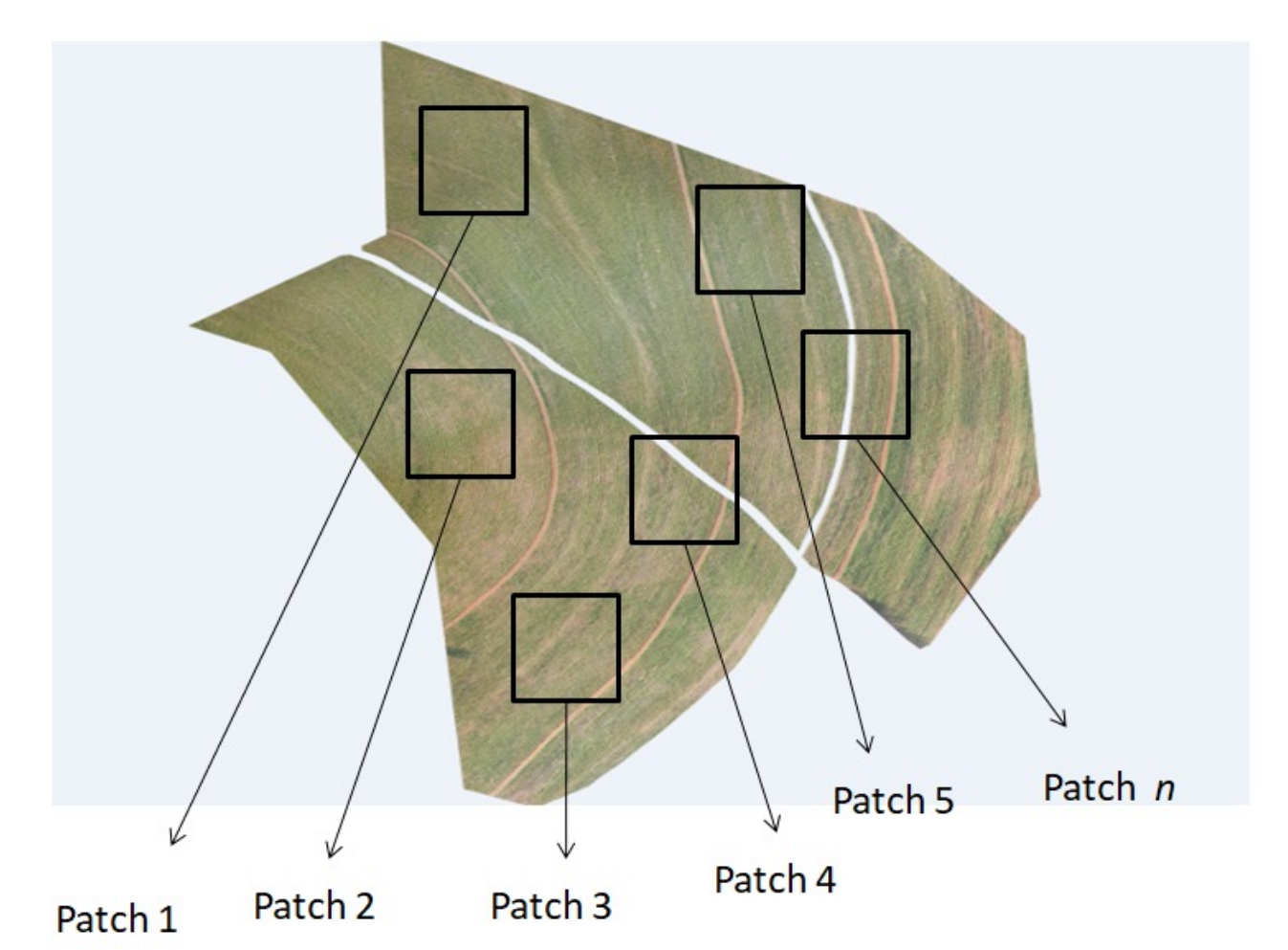


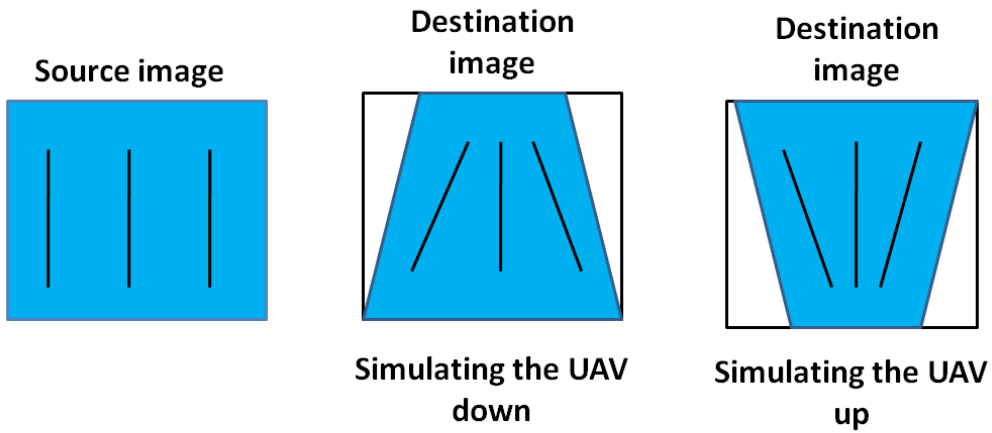
Figure: For the construction of both datasets, we use the same strategy, which consists of making cutouts in a mosaic to extract the patches.

Linear Dataset

- We considered only three linear distortions that may occur during flight: translation, rotation and perspective transformation.
- **Rotation:** we used $\theta = \{0°, 5°, 10°, 15°\}$.
- **Translation:** images were translated by 25 pixels in 4 possible directions: right and top; right and down; left and top; and left and down, and the original image.

Linear Deformation Dataset

- **Perspective transformation:** it simulates UAV up and down possibilities in moments of image capture.
  - We keep the proportion of lines identical to the original image.
  - For the columns, the proportions in each of the distorted classes created were: (0.05, 0.66); (0.05, 0.77); (0.02, 0.66); (0.02, 0.77).



Linear Deformation Dataset

- We cropped a 64×64 pixels region aligned with the center of the image to remove black areas that could influence the neural network training and testing.

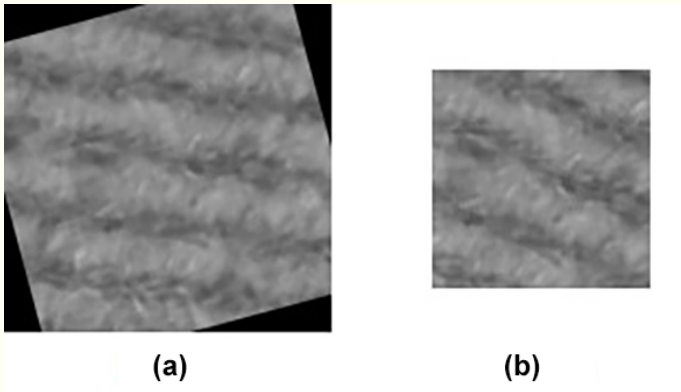


Figure: (a) Image after a 15-degree rotation transformation; (b) Cropped region with 64×64 pixels size.

Linear Deformation Dataset

- We trained the network with the subtraction of the deformed patches (A) and their deformation-free patches (B), to relate them.
- To avoid negative values, we normalized the computed x_{ij} values as follows:

$$x_{ij} = max(b_{ij} - a_{ij}, 0) \tag{1}$$

- Where $a_{ij} \in A$ represents a pixel of patch A, $b_{ij} \in B$ represents a pixel of patch B and $x_{ij} \in X$ represents a pixel of patch X.

Linear Deformation Dataset

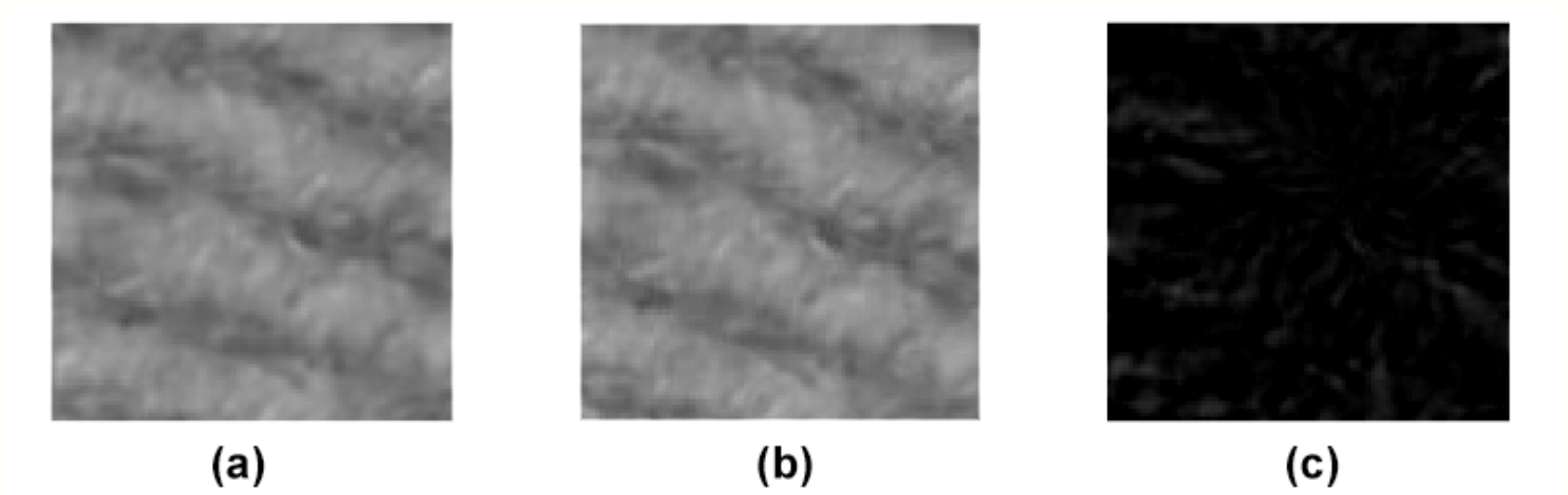


Figure: (a) Artificially distorted image; (b) Corresponding distortion-free image; (c) Result from the subtraction operation.

Non-Linear Dataset

- We use the work of Eppenhof et al. (2019) to create non-linear deformations.
- We concatenate other linear transformations through an interpolator method, in order to have as many random transformations as possible.
- We cropped a 64×64 pixels region aligned with the center of the image to remove black areas that could influence the neural network training and testing, such as Linear Dataset.

Non-Linear Dataset

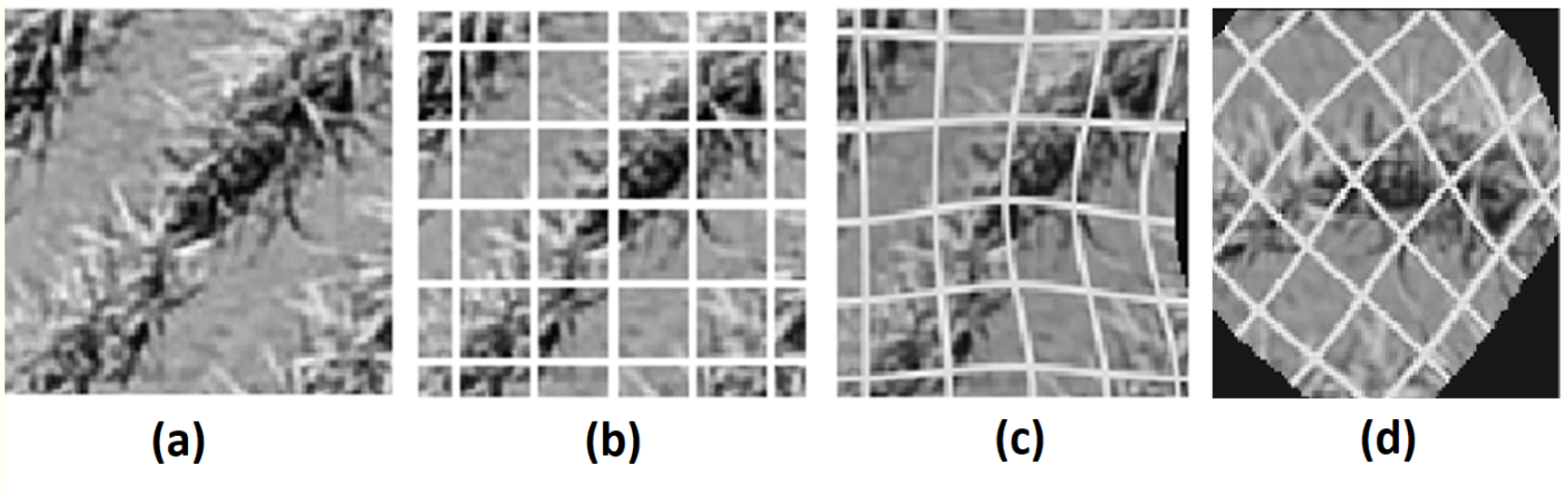


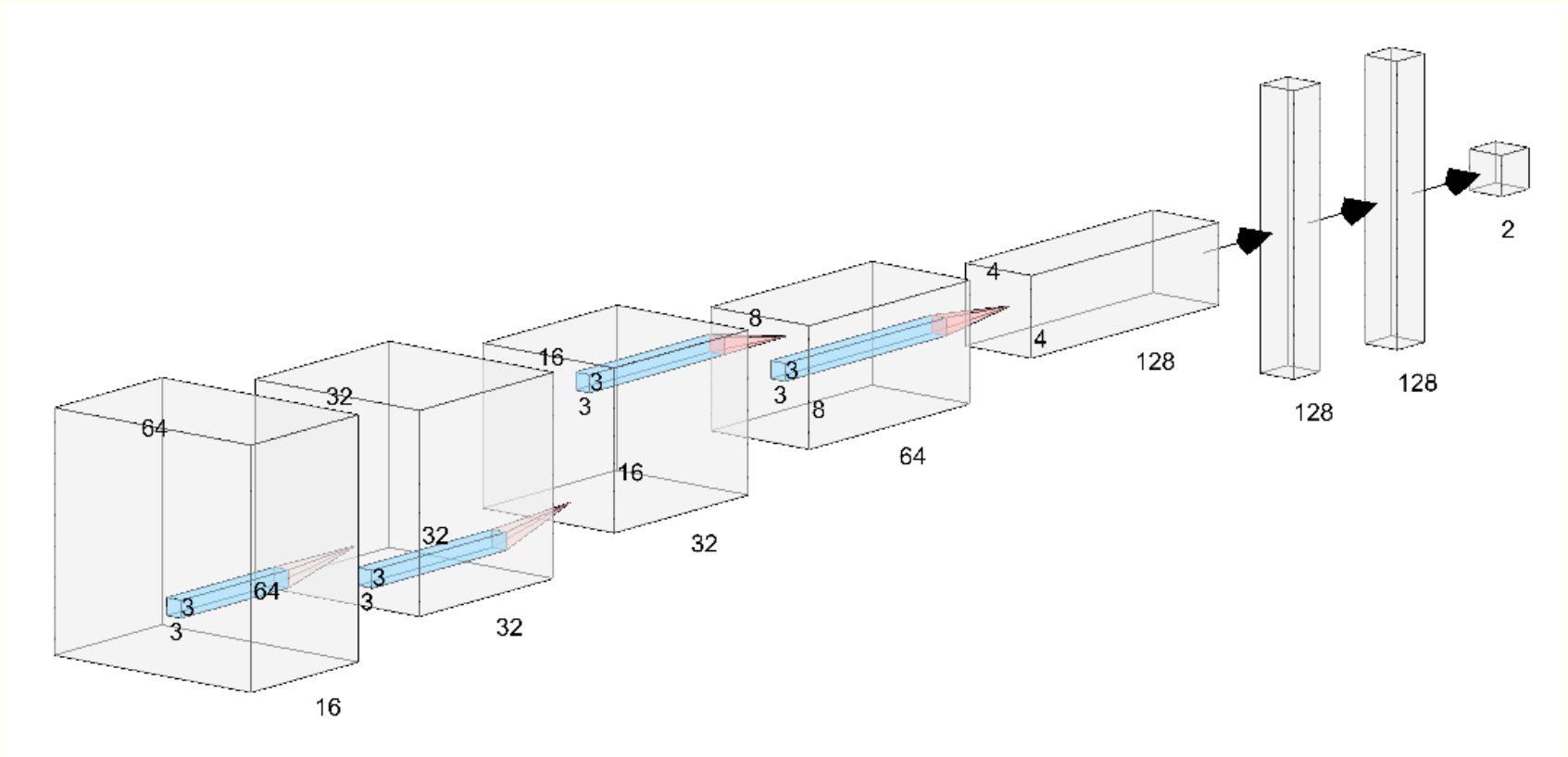
Figure: Distortion example.

Experiments and Results

- To implement the CNN we used the Python version of Tensorflow, an open-source library developed by Google.
- We evaluated our CNN model using both datasets.
- For each dataset we selected 75% of the samples to compose the training set, while the remaining images were used for validation.

Experiments and Results

- We proposed an alternative architecture inspired by AlexNet, which has a lower computational cost.



Experiments and Results

- In order to improve the evaluation of our CNN model we compared its results with the ones obtained by 4 traditional CNN models:
  - InceptionV3
  - ResNet
  - SqueezeNet
  - VGG-16

Experiments and Results

- For this comparison we used pre-trained networks on the 2012 ImageNet dataset and fine-tuned the whole CNN to our classification problem for 20 epochs.
- These networks have larger input sizes than our samples, so the images were enlarged for adjustment.

Experiments and Results

- Results show that CNNs can identify the presence of linear deformations.

CNN model	Translation		Rotation		Perspective	
	DS1	DS2	DS1	DS2	DS1	DS2
ResNet	91.83	48.13	95.00	96.84	59.55	62.63
InceptionV3	20.00	60.10	98.48	98.23	20.00	65.96
VGG-16	94.76	65.15	98.63	98.74	84.89	75.20
SqueezeNet	90.51	40.40	91.77	96.15	55.68	55.20
Proposed	**96.92**	**70.24**	**99.85**	**99.18**	**95.50**	**91.47**

Table: Results (%) obtained for our CNN and the compared ones.

Experiments and Results

- Results show that CNNs can identify the presence of non-linear deformations.

CNN model	DS1	DS2
ResNet	98.95	94.36
InceptionV3	**99.25**	**98.84**
VGG-16	98.88	95.35
SqueezeNet	98.43	96.41
Proposed	97.53	92.92

Table: Results (%) obtained for each CNN model for non-linear deformations.

Experiments and Results

- With a small loss of precision compared to traditional CNNs to address non-linear deformations, the architecture proposed in this work has the advantage of having a considerably lower computational cost.

CNN model	Number of parameters
ResNet	22,591,810
InceptionV3	22,081,826
VGG-16	14,797,122
SqueezeNet	723,522
Proposed	141,058

Table: Number of parameters of each CNN model.

Experiments and Results

- The results show the ability of CNNs to correctly identify possible deformations inherent to the flight of a UAV, whether these deformations are of a linear or non-linear nature.
- Also, an architecture with lower computational cost, but with still high accuracy, proved to be effective in treating the problem.
- This information would be extremely valid for the subsequent deformation correction process.

Deformable Co-Registration Using Deep Learning

Image Dataset

- We considered two dataset to evaluate the approach proposed (soybean and cotton).
- Both datasets have 1280×960 pixels size per image and an average of 80% overlap between images.
- The spectra present in the images are Blue, Green, Red, NIR, and Red Edge.

Image Dataset

- Images were obtained in a single flight, without any type of pre-processing.
- This flight took place at an average height of 100 meters, at an average speed of $20m/s$.
  - Under these conditions, the GSD is $6.8cm/pixel$.

Image Dataset

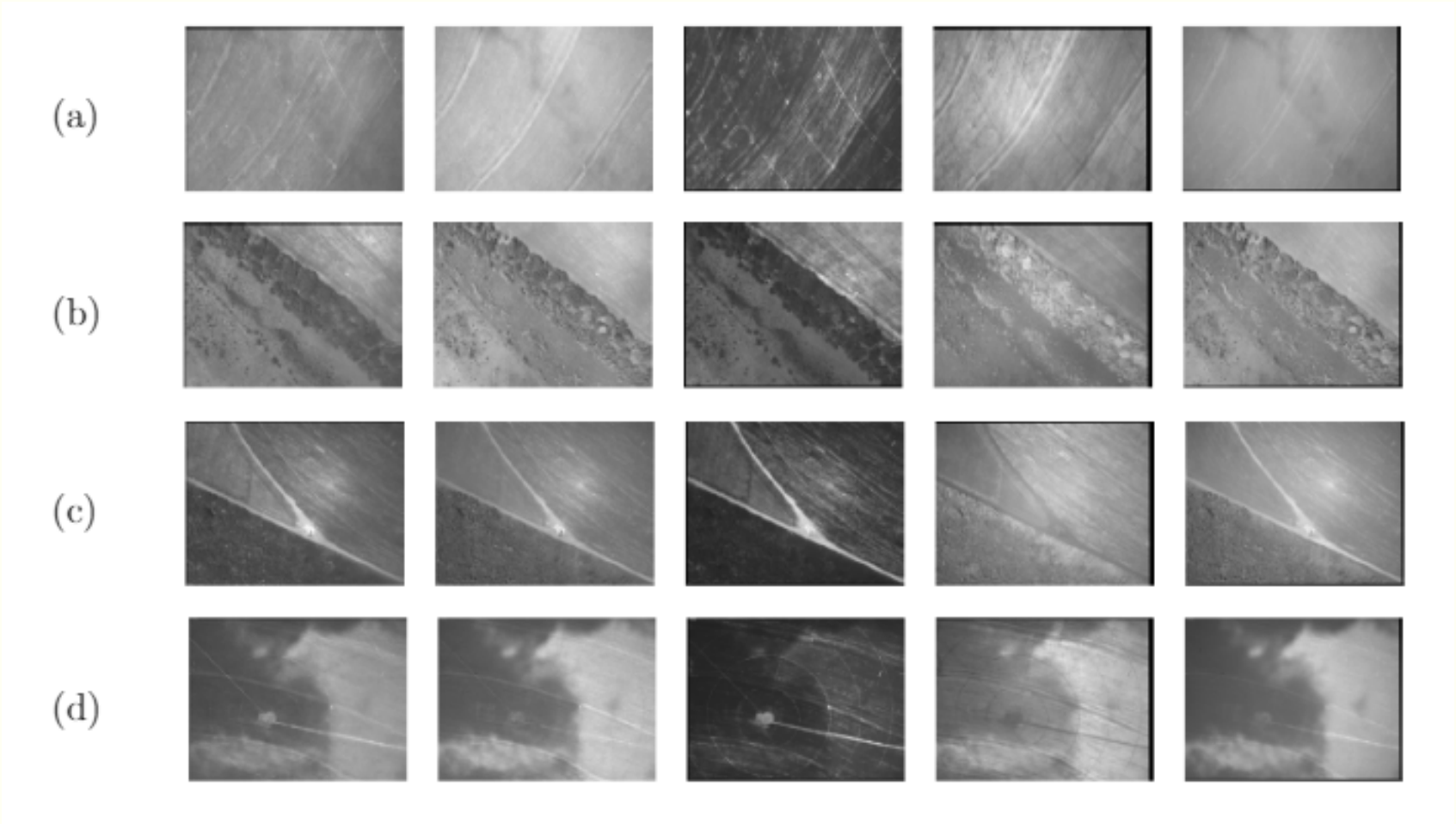


Figure: Examples of images from soybean dataset: (a), (b), (c), and (d) present, respectively, four different scenes captured by UAV, with the columns represent the respective bands of the scene.

Image Dataset

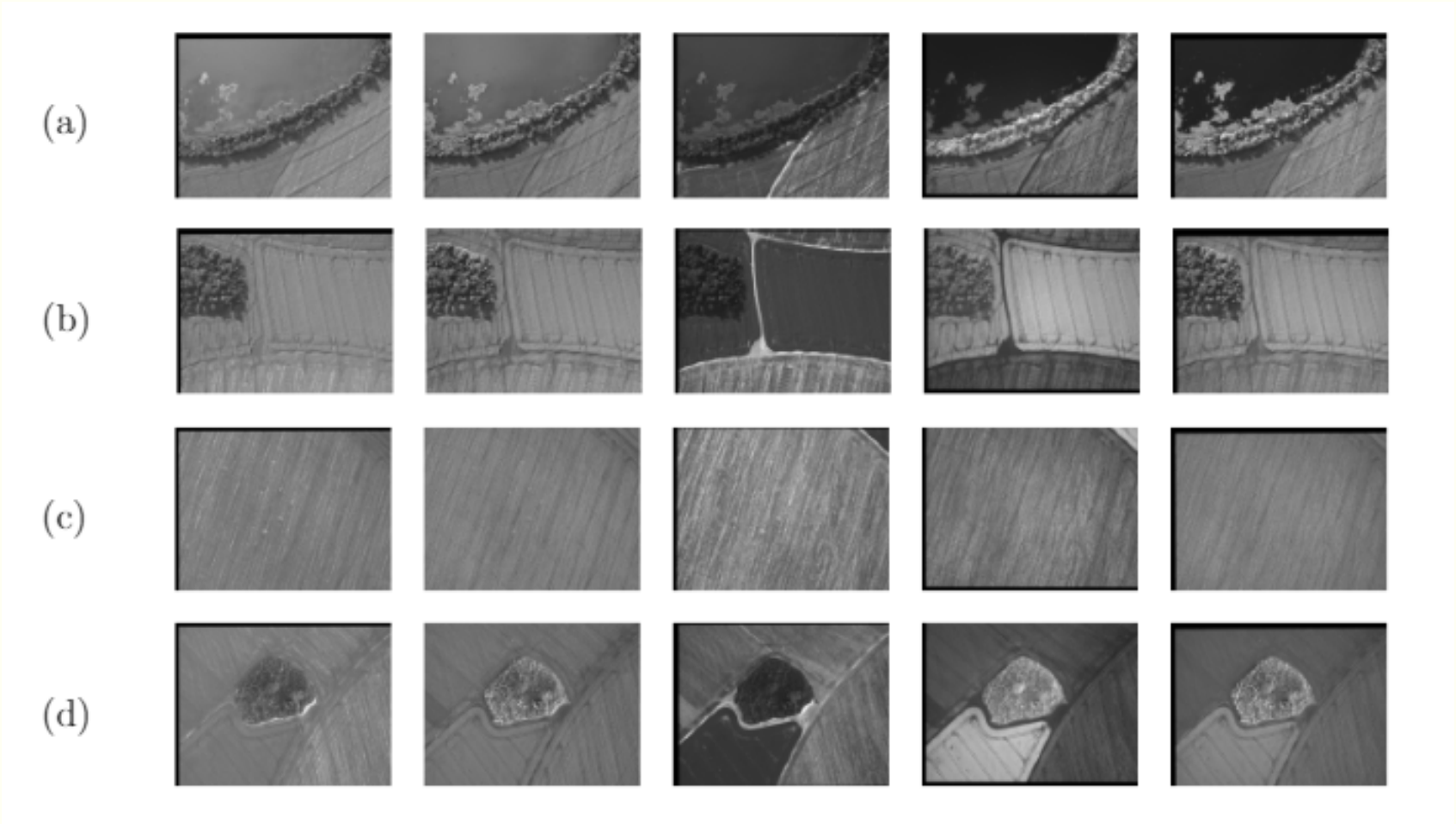


Figure: Examples of images from cotton dataset: (a), (b), (c), and (d) present, respectively, four different scenes captured by UAV, with the columns represent the respective bands of the scene.

Data Preparation

- The images were manually marked by two different experts, to have a ground truth.
- The experts marked 12 points in the green band.
- In the sequence, the experts marked the equivalent points in the other bands.

Data Preparation

	Blue	Green	Red	NIR	Red Edge
Blue	-	5.18	15.75	15.07	12.14
Green	5.18	-	15.09	12.33	4.02
Red	15.75	15.09	-	29.25	14.79
NIR	15.07	12.33	29.25	-	16.17
Red Edge	12.14	4.02	14.79	16.17	-

Table: Misalignment average, in pixels, between the sensors present in soybean dataset.

Data Preparation

	Blue	Green	Red	NIR	Red Edge
Blue	-	28.30	12.11	33.28	8.48
Green	28.30	-	21.44	24.01	35.66
Red	12.11	21.44	-	14.94	21.30
NIR	33.28	24.01	14.94	-	39.37
Red Edge	8.48	35.66	21.30	39.37	-

Table: Misalignment average, in pixels, between the sensors present in cotton dataset.

Data Preparation

- We have proportionally reduced the images to 20% of their original size, preserving their content and proportionality.
- The purpose of this operation is to reduce the computational cost of further training in relation to the original size of the images.
- The images will be 256×192 pixels size.

Data Preparation

- The non-linear deformations consisted of two random grids, one for displacements in the *y-directions*, and the other for displacements in the *x-directions*.
- The transformation is defined through a 3×3 grid point B-spline grid, where random displacements are ranging from a uniform distribution.

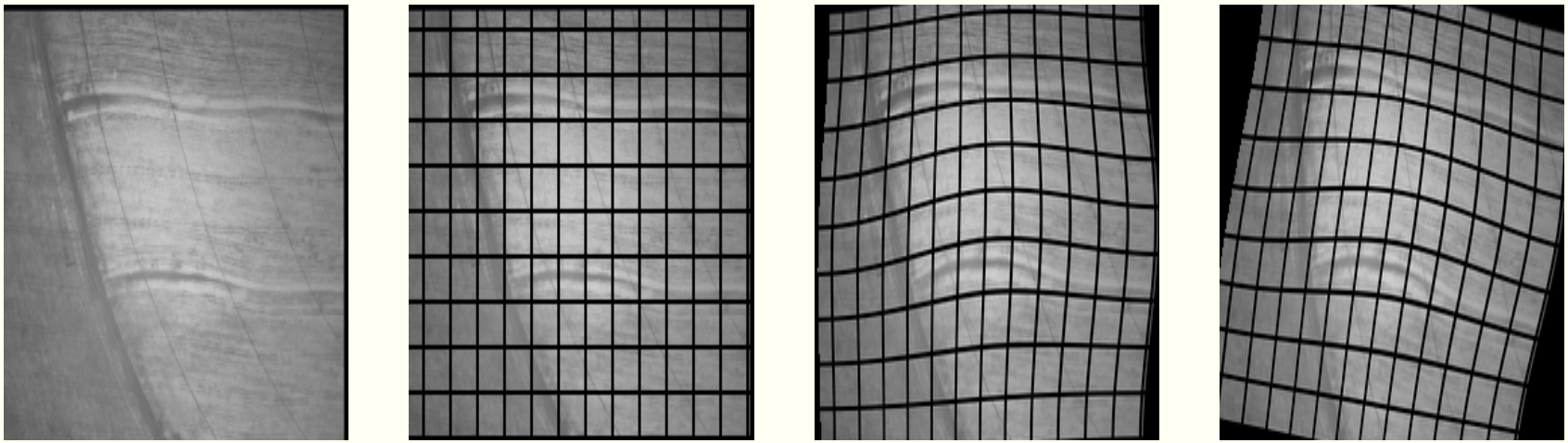
Grid point displacement		
Grid size	x	y
3×3	$[-0.05, 0.05]$	$[-0.05, 0.05]$

Table: Parameters for B-spline transformations.

Data Preparation

- We also inserted two more rigid deformations often found in an UAV flight: rotation and scale.
- For the scale, we consider variations of $\pm 2\%$, with the original.
- For the rotation, we consider a variation of up to 10 degrees around the point $(0.5, 0.5)$.

Data Preparation



(a) (b) (c) (d)

Figure: Example of distortion in near-infrared band.

Data Preparation

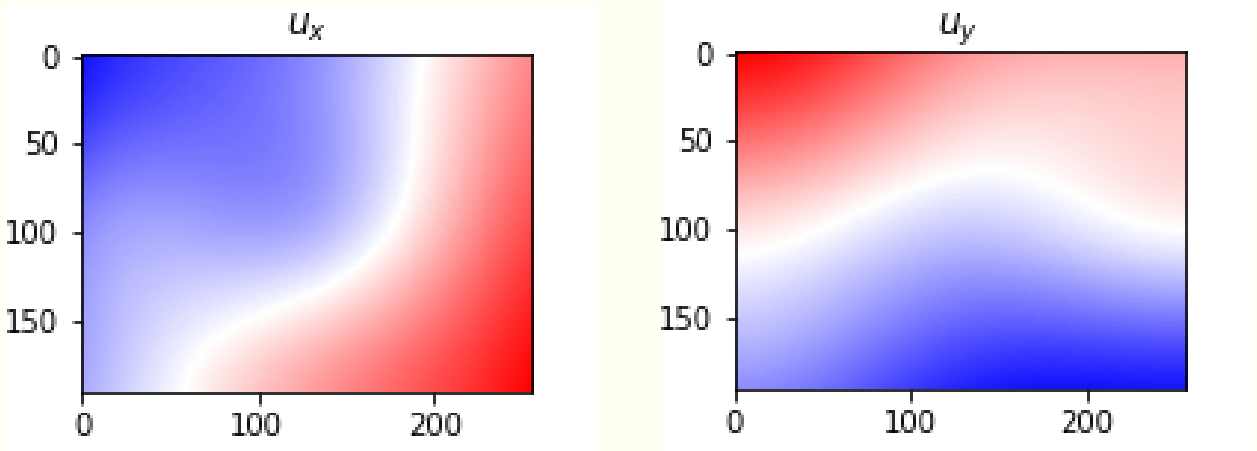
- The aim will be to learn the displacement field (U) of the moving image to the fixed image. The displacement field (U) is calculated as follows:

$$U = originalGrid - transformedGrid \tag{2}$$

- Where *originalGrid* refers to the deformation-free image grid and *transfomedGrid* refers to the image grid after inserting the deformations.

Data Preparation

- There is the visual presentation of the displacement fields, with the respective color map.



(a) (b)

Figure: Example of displacement fields: in (a) we have the displacement field for the x axis; in (b) we have the displacement field for the y axis. The colors blue and red represent positive and negative displacements respectively.

End-to-end training

- The network is trained to maximize the accuracy of the estimate between the estimated vector field ($\hat{U}$) and the true vector field (U).
- We used Adam optimizer with a learning rate of 10^{-4}.
- In addition, the loss is defined as:

$$loss = \frac{1}{n}\sum_{j=1}^{n}|u_j - \hat{u}_j| \tag{3}$$

- Where n refers to number of iterations.

End-to-end training

- The end-to-end architecture used is the fully-convolutional neural network proposed by Ronneberger, Fischer e Brox (2015).
- This architecture is known as U-Net and its use is mainly focused on the segmentation of 2D and 3D medical images.

End-to-end training

- To adapt the U-Net to the registration problem, the input layer has two channels, one considered the fixed band and the other for the moved band.
- The output layer is also adapted to have two channels, where each channel corresponds to the form vector field on one of the axes (x and y).
- Activation functions are the Leaky ReLU parameterized as $\phi(x) = max(x, 0.01x)$

End-to-end training

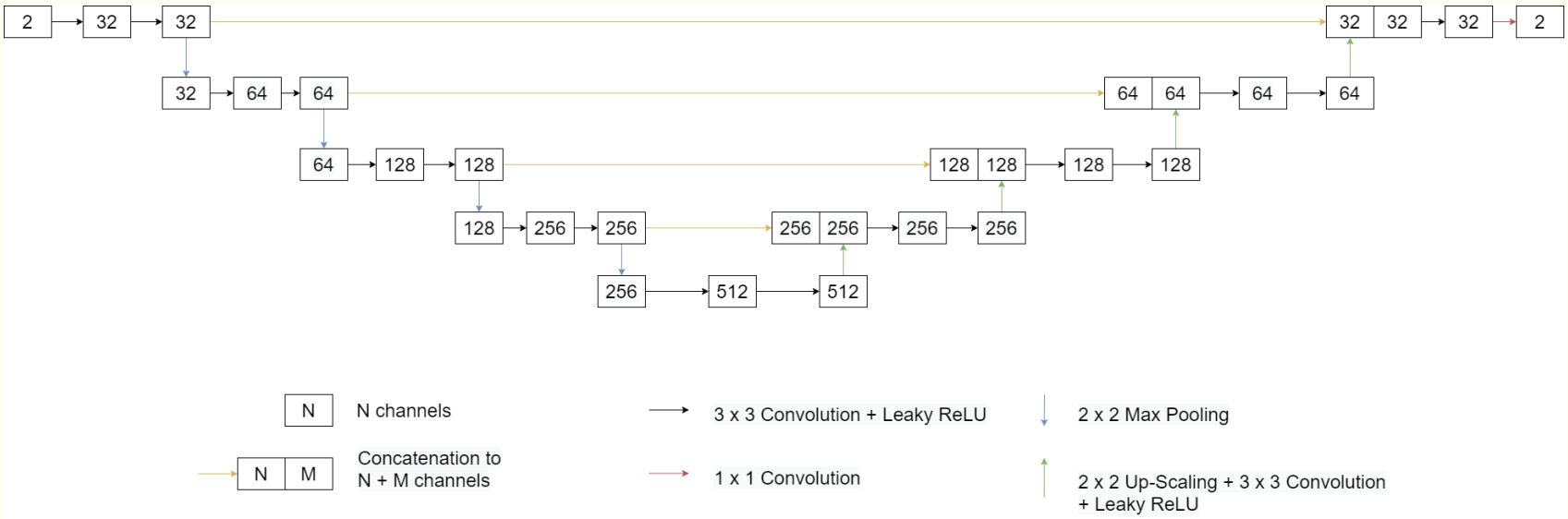


Figure: The proposed network architecture. The network takes two bands as input, and outputs two maps: one for each vector field axis.

Experiments and Results

- First, we tested it on a set of images in gray scales.
- Each image was deformed under the same conditions in multiespectral images, however, the images are 128×128 pixels size.
- The dataset has a total of 328 images and was divided into sets of training, validation, and tests, in the proportion of 70%, 15%, and 15% respectively.

Experiments and Results

- In this scenario, the network input consists of a two-channel image:
  - The first channel contains the deformation-free image.
  - The second channel contains the deformed image.
- The training was carried out with 500 epochs and obtained an accuracy of **97.54%**.

Experiments and Results

- We apply the transformation learned by the network, interpolating them in a new grid, as follows:

$$I_R(x, y) = I_M(x + U(x, y), y + V(x, y)) \tag{4}$$

- Where I_R is the registered band, I_M is the moved band and $U(x, y)$ and $V(x, y)$ are the predicted vector fields for the x and y axes respectively.

Experiments and Results

- With the displacement field predicted by the network applied to the deformed image, we can visualize the result through an overlap between the two images.

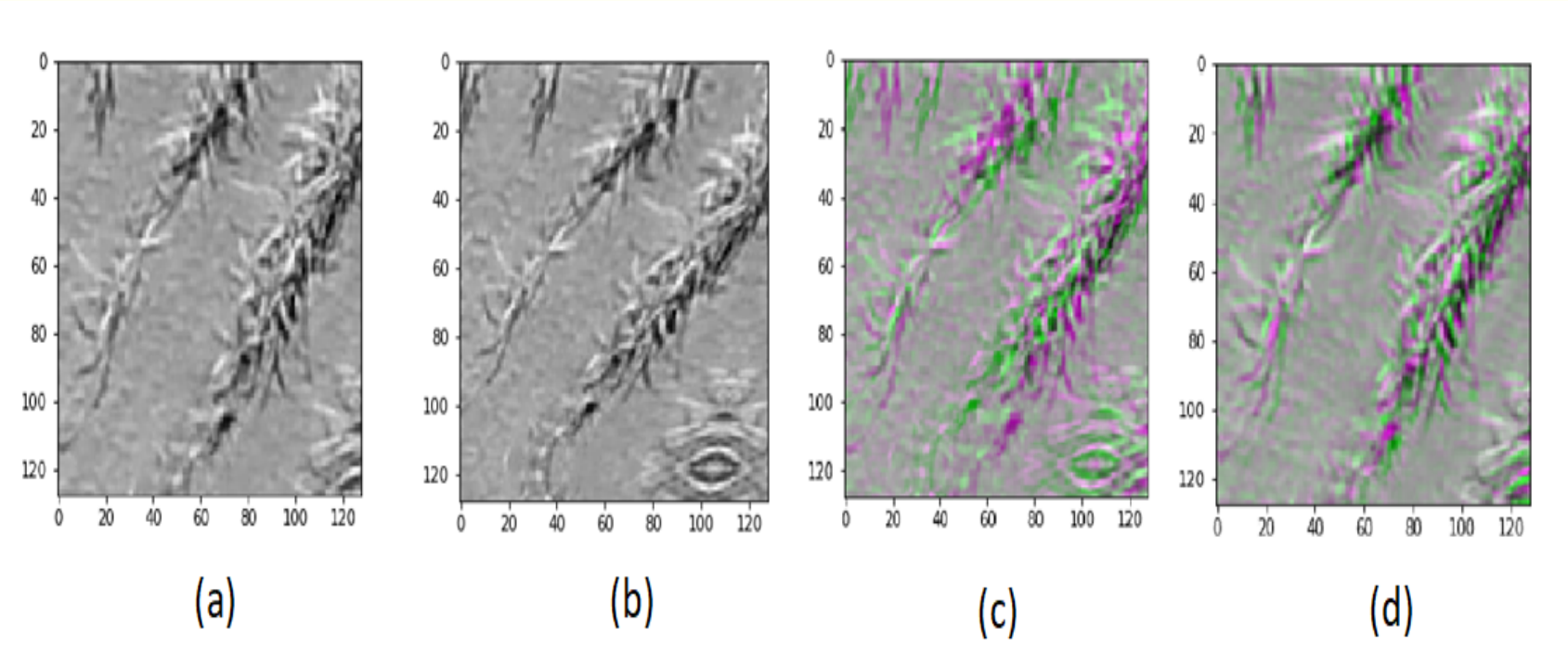


Figure: Example of registration in RGB image from the field predicted by our network.

Experiments and Results

- As our network proved to be able to learn the displacement field in gray scales images, we can evaluate the approach proposed in multispectral datasets (soybean and cotton).
- Both datasets were divided into sets of training, validation, and test, in the proportion of 70%, 15%, and 15% respectively.

Experiments and Results

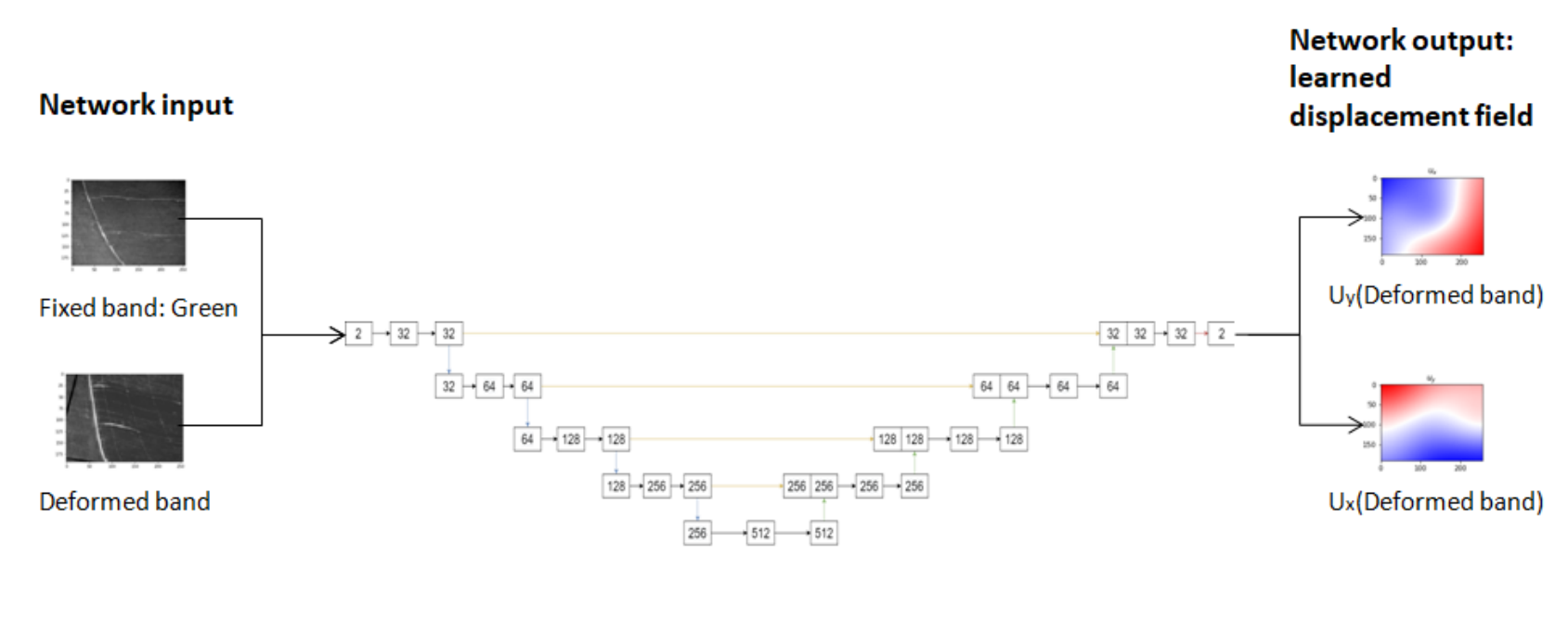


Figure: The proposed approach: the input consists of an image with two channels (a fixed band and another moved band), and the output consists of two displacement fields (in x and y) of the band moved to the fixed band.

Experiments and Results

- 2000 training epochs were considered for this scenario.
- For the soybean dataset, we obtained an accuracy ranging from 89.90% to 93.79%
- For the cotton dataset, we obtained an accuracy ranging from 90.01% to 91.21%

Experiments and Results

Deformed Band	Accuracy	
	Soybean Dataset	Cotton Dataset
Blue	89.90%	90.01%
Red	89.95%	90.11%
NIR	90.50%	91.21%
Red Edge	93.79%	90.54%

Table: Summary of accuracy for each of the moved bands considered.

Experiments and Results

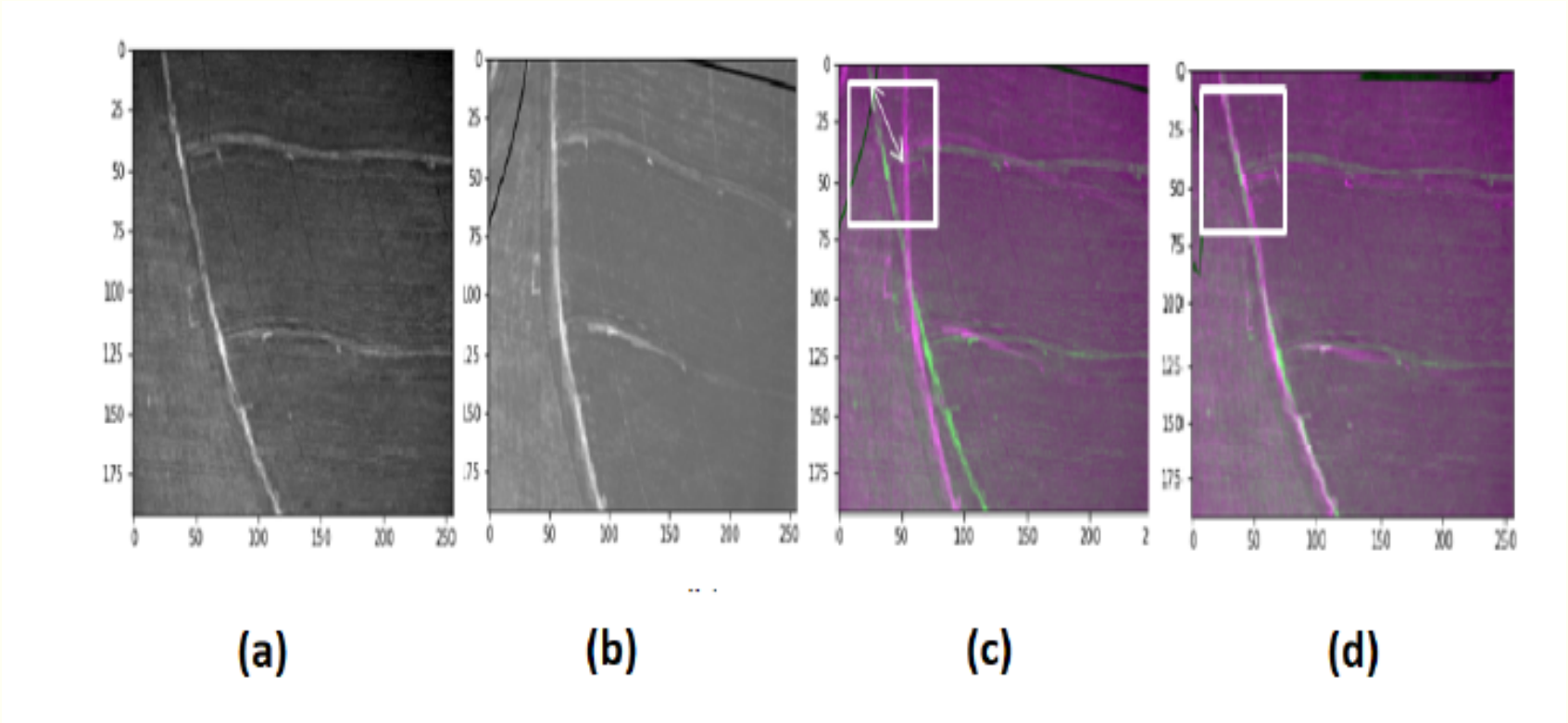


Figure: Example of co-registration in soybean dataset.

Experiments and Results

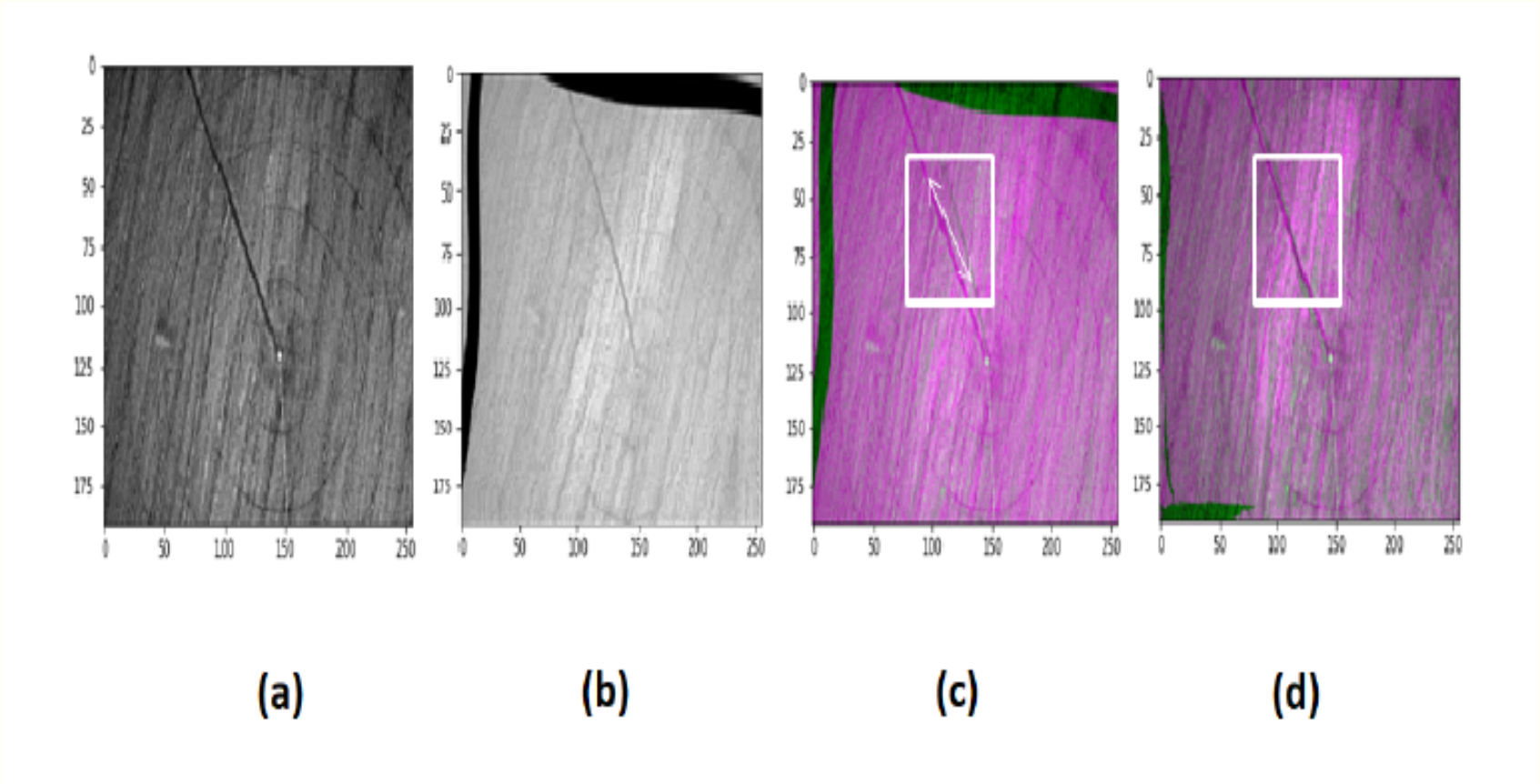


Figure: Example of co-registration in cotton dataset.

Discussions

- The proposed approach proved to be capable of learning a deformation field between bands of multispectral images with considerable precision.
- Our network is able to perform co-registration after training without the need for the manual marking of points.
- The multispectral factor should be considered as an additional difficulty.

Discussions

- Although our proposal is related to that of Eppenhof et al. (2019), we take into account multispectral images.
- In addition to the content of each one of the images in our training set being totally different, while in Eppenhof et al. (2019) the images are exclusive of lungs captured through Computed Tomography (CT).

Discussions

- In Dias Junior et al. (2020), the co-registration between the same bands considered in this work is carried out.
- However, Dias Junior et al. (2020) did not consider the presence of non-linear deformations.
- Besides needing the manual marking of points for the co-registration task.

Conclusions

Main contributions

- Demonstration of the ability of CNNs to identify the presence (or not) of linear and non-linear deformations.
- Proposition of a framework based on deep learning for co-registration between bands of a multispectral image, which is capable of treating linear and non-linear deformations, without the need for manual point marking.

Limitations

- This work considered for the identification and correction of deformations a random set of deformations and their combinations. These deformations may not represent a real scenario.
- Also, our analysis of co-registration was visual. Therefore, we chose to resize the images, preserving their proportions, however, this can invariably lead to the loss of information from the images.

Future Works

- Consider other deformations (and their variations) in images obtained by UAVs.
- As the green band was always considered "fixed" in this work, it is intended to define a graph to perform the co-registration, so that the best matches are found between the bands.
- Evaluate the performance of the proposed framework with images obtained by other sensors than those considered in this work.
- Analyze quantitatively the co-registration results obtained by our proposal.

Acknowledgements

Thank You!