Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

Crossing points detection in plain weave for old paintings with deep learning☆

A. Delgado [a], Laura Alba-Carcelén [b], Juan J. Murillo-Fuentes [a],*

[a] *Dep. Teoría de la Señal y Comunicaciones. ETSI. Universidad de Sevilla., Camino de los Descubrimientos sn, Sevilla, 41092, Spain*
[b] *Museo Nacional del Prado, Madrid, Spain*

ARTICLE INFO

ABSTRACT

In the forensic studies of painting masterpieces, the analysis of the support is of major importance. For plain weave fabrics, the densities of vertical and horizontal threads are used as main features, while angle deviations from the vertical and horizontal axis are also of help. These features can be studied locally through the canvas. In this work, deep learning is proposed as a tool to perform these local densities and angle studies. We trained the model with samples from 36 paintings by Velázquez, Rubens or Ribera, among others. The data preparation and augmentation are dealt with at a first stage of the pipeline. We then focus on the supervised segmentation of crossing points between threads. The U-Net with inception and Dice loss are presented as good choices for this task. Densities and angles are then estimated based on the segmented crossing points. We report test results of the analysis of a few canvases and a comparison with methods in the frequency domain, widely used in this problem. We concluded that this new approach successes in some cases where the frequency analysis tools fail, while improves the results in others. Besides, our proposal does not need the labeling of part of the to be processed image. As case studies, we apply this novel algorithm to the analysis of two pairs of canvases by Velázquez and Murillo, to conclude that the fabrics used came from the same roll.

1. Introduction

1.1. Fabric analysis and plain weave

The weave and patterns on the fabric of paintings can be seen as features or fingerprints that help in the forensic analysis of paintings, e.g., to date and assign authorship (Alba and Murillo-Fuentes, 2021). The type of weave, the material, the number of threads per centimeter or the weight of fabrics have been widely studied (de Carbonnel, 1980). A photograph of the back of the painting can be processed to study the fabric, but in many cases the canvas has been relined to reinforce it and the original fabric cannot be observed directly. Instead, X-ray plates of the paintings are usually analyzed. Radiographs are much harder to process because the frame and the painting itself, including cracks, are observed as noise.

Plain weave fabrics[1] have been widely used as the support of paintings. In plain weave we have horizontal and vertical threads

intertwined. In the loom, a set of threads, the warp, are parallel arranged from back to front, while another thread, the weft, is passed iteratively from side to side, i.e., orthogonally to the warp. At this point it is interesting to remark that separations between threads in the warp presents a deterministic pattern that depends on how the threads have been arranged, while in the weft the separation along the roll depends on how the threads have been tightened. This is the reason why the thread counting, i.e., the count of threads per cm, both in the vertical and horizontal axes, has been widely used as characteristic of the canvases. Besides, because of deformations of the fabric around nails in the stretcher persists in time,[2] deviations of threads respect to the horizontal and vertical axis are useful for the curator to study the painting and provide information on transformations of areas in the canvas, checking for integrity, and better understanding how the painter's workshop was organized in the productions of series.

[1] Also known as tabby, calico or tafetta weave.
[2] Prime is applied to the cloth after nailing, and this prime acts as a glue.
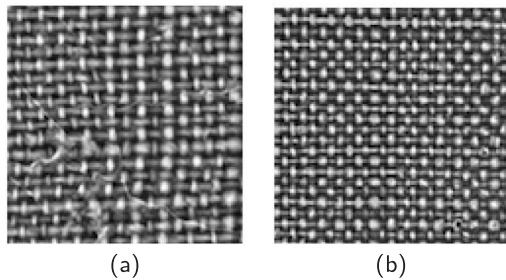
Fig. 1. Scenarios where the FT fails, fabrics in canvases by (a) Rubens with different thread widths and (b) Velázquez where vertical threads are seen as widenings of horizontal ones at the crossing points.

The FT has been successfully applied to this problem of plain wave canvases analysis (Johnson and Erdmann, 2013; Simois and Murillo-Fuentes, 2018), exhibiting quite a robust performance. However, we found that the FT fails in two common scenarios. First, whenever threads are of different widths, for the pattern is not uniform, the FT detects several maxima. As a result, in nearby areas of the canvas the FT provides quite different threads densities, since it changes from one maximum to another as we process locations one next to the other. In Fig. 1.(a) we include a sample of the X-ray of the canvas Adan and Eva from Rubens (1628), where threads of several different widths are observed. On the other hand, in some fabrics and in one direction, usually the warp, the threads are quite tight and the orthogonal ones are just perceived as widenings at the crossing points. In this scenario the FT is unable to provide any estimation of the thread density. In Fig. 1.(b) we bring a sample of the X-ray plate of the Prince Baltasar Carlos on Horseback by Velázquez (P001180 in MNP inventory) (de Silva y Velázquez, 1634), horizontal threads are easily observed while the vertical ones can be only identified by focusing on the widening of the horizontal threads.

1.2. Pipeline and contributions

In the following, after describing related work in Section 2, we face the analysis of thread densities and angles in plain weaves by means of segmentation. As a first solution, we tried to segment vertical and horizontal threads independently. However, as fabrics have severe rotations at some points, the model was unable to distinguish a vertical (or horizontal) thread from a severely rotated horizontal (vertical) one. As a result, the learning did not converge and we decided to segment (Minaee et al., 2022) crossing points instead, with just one model. Accordingly, the pipeline was designed as follows.

In the dataset generation, described in Section 3, we used paintings from several authors in the 17th-19th centuries. From these paintings, random samples were obtained and labeled. We annotated both the vertical and horizontal threads in every sample. The labeled samples were divided into training, validation and test subsets. We investigated several models, see Section 4, and adopted the U-Net with inception (Szegedy et al., 2015). The Dice loss function was chosen, exhibiting good performance. At this point, it is interesting to note that we do not have densities nor angle estimations, but the locations of the crossing points. Hence, an algorithm to estimate these values from the crossing points was developed. This approach is described in Section 5. The results of the training of the proposed model and its validation are included in Section 6. In production, we process a whole canvas to provide the densities and angle deviations at every point. Note that we are not using the trained model as transfer learning to analyze a new canvas, as this would involve a labeling process for every new painting (Aradillas et al., 2021). We do not assume any annotation for a new canvas analysis (Maaten and Erdmann, 2015). Section 7 is devoted to illustrating the good performance of the developed approach

compared to the method in the frequency domain. A couple of pairs of canvases by Velázquez and Murillo are analyzed. We also include the study of an X-ray plate of a painting by Ribera. We end with conclusions.

The main contributions of the paper are as follows:

1. A description of problems of the FT applied to the thread counting problem.
2. A method to crop and label X-rays of paintings, augmenting the resulting dataset.
3. Four DL models to detect crossing points of threads in plain weave, working in the spatial domain. These DL models follow the U-Net architecture and have been programmed using Keras-TensorFlow. We report that the segmentation of vertical, or horizontal, threads yields poor results.
4. An approach to compute threads densities and angle estimations from crossing points.
5. A whole thread counting algorithm in the spatial domain with no need of labeling part of the to be processed canvas.
6. Analysis of the errors in the estimation of thread densities for labeled data of X-rays of different qualities and fabrics of different densities.
7. Application to a pair of case studies with canvases by Velázquez and Murillo, to conclude the fabrics used coming from the same roll, very much outperforming the FT approach.

2. Related work

Image processing has been applied to the study of priceless paintings (Barni et al., 2005; Cornelis et al., 2017; Deligiannis et al., 2017; Johnson et al., 2008). Recently, algorithms in the machine learning (ML) field based on artificial neural networks have been applied. In Rucoba-Calderón et al. (2022) crack detection is solved by applying K-SVD and in Sizyakin et al. (2020) convolutional neural networks (CNN) were used. CNN were also applied to automatic classification of paintings (Roberto et al., 2020). In Pu et al. (2020) auto-encoders (AE) were used for image separation. In Zou et al. (2021) deep learning (DL) was applied for virtual restoration of colored paintings. Authentication and forgery detection has also been the focus of these techniques (Polatkan et al., 2009; Nemade et al., 2017). Segmentation approaches based on U-Net (Ronneberger et al., 2015) and AE (Rumelhart et al., 1986; Goodfellow et al., 2016) were applied to image restoration by inpainting.

Within the field of image processing and machine learning (ML) the study of the fabric has received particular attention. In Escofet et al. (2001) the theoretical frequency analysis of fabrics was introduced, being updated in Simois and Murillo-Fuentes (2018). Later, this theoretical background was translated into studies of canvases (Johnson et al., 2010; Johnson and Erdmann, 2013) by means of the Fourier transform (FT). Another form of transform was also tried (Yang et al., 2015) and the power spectral density was introduced as a feature of the canvas in Simois and Murillo-Fuentes (2018). These transform-based tools are unsupervised, as no previous labeling of the threads in the images or their densities was needed. These FT approaches are robust and useful to process many masterpieces, but they fail in some scenarios, as discussed earlier. In Maaten and Erdmann (2015) the authors presented a Bayesian tool to predict the positions of the crossing points in plain weave. This is, up to our knowledge, the only ML tool applied to the thread counting problem. However, a prior labeling stage was needed for the canvas at hand, where the curator needs to mark the crossing points for a given area of the to be processed canvas. In this manuscript we present a novel ML tool for plain weave thread counting to solve some drawbacks of the FT approaches, with no need of pre-labeling.
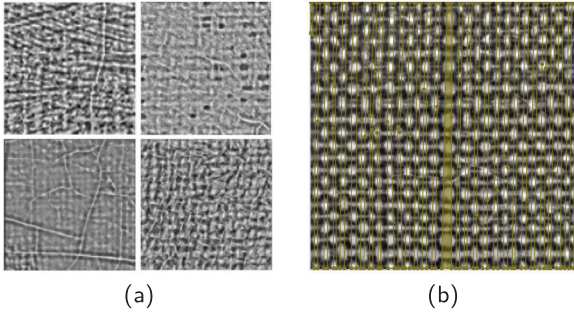
Fig. 2. Examples of (a) four samples where the fabric is hardly observed and (b) a labeled crop with annotations for vertical threads, one of the labeled threads is highlighted.

3. Data preparation

3.1. Dataset

We first selected 36 paintings from the Museo Nacional del Prado (MNP), of more than 13 different painters. Canvases from Rubens, Velázquez, Lorena, Swanevelt, Dughet, Poussin, Both, Lemaire, or Ribera were included in the dataset. These paintings were selected to be representative of several densities in the range 6 to 23 threads per cm (thr/cm), different resolutions of the image and noise conditions. This encompasses the usual densities found in canvases. For example, in the French paintings analyzed for the 17th–20th centuries and 44 painters the densities of the threads were in the range 6-23.3 thr/cm. For the 17th and 18th centuries, the range reduces to 8.6-20 thr/cm (de Carbonnel, 1980).

The images of the X-ray plates were obtained after digitalization, with resolutions ranging from 80 to 200 pixels per cm (ppc). The images were enhanced with algorithms based on their local mean and variance (Lee, 1980) then scaled, see Appendix for a description of these steps. For every plate we cropped 40 random areas of 1.5×1.5 cm and resampled them, if needed, to 200 ppc. For every canvas, approximately 7 samples out of the 40 were labeled. This number was reduced or increased if its densities and quality were already represented or not by other similar fabrics in the dataset. Overall, we labeled 239 samples. The labeling of these samples was performed not for the crossing points but for the vertical and horizontal threads independently, then a unique labeled image was obtained with the crossing points. This stage was specially elaborated as in many cases the fabric was hardly observed in the X-ray plate, see four different samples in Fig. 2.(a). For high densities, the labeling becomes challenging as the space between threads is quite narrow, if any. See Fig. 2.(b), where the annotations for one sample is included and one of the annotations is highlighted. More than one person participated in the labeling. Therefore, for the same canvas we may have slightly different widths for the annotations.

Then, we divided the dataset into training, validation and test. Instead of randomly distributing the samples into these three groups, we took the labeled samples from 4 paintings for validation, 31 samples overall, and the samples of 3 other paintings for test purposes, with 18 labeled samples. Hence, out of the available paintings, 29 were used for training, 4 for validation and 3 for test. We enforced that not only different paintings were used in each group, but that canvases with labeled fabrics known to come from the same roll were placed in the same subset. Paintings in the validation and test datasets were selected to be representative of different densities and qualities. At this point, it is important to remark that these are not the final samples used. Next, we explain how to generate the samples fed to the model for the training, validation and test stages.

3.2. Data generation and augmentation

We need a large dataset to represent the different possible inputs. In our design, we labeled 1.5×1.5 cm samples at 200 ppc resolution. However, the model, as explained later, is fed with 1×1 cm images (200×200 pixels). The first step to generate the input datasets was to obtain these images from every sample. We performed this task by first taking the corners of the sample to get the first 4 images. Then four rotated versions of two images given by the (50:250, 50:250) and (65:265, 65:265) pixels of the sample were added, with random angles in the ranges $[2, 7]$, $[-2, -7]$, $[8, 12]$, $[-8, -12]$. We also added two rotated versions of the images given by the (65:265, 65:265) pixels of the sample, with random angles in the ranges $[2, 7]$, $[-2, -7]$. This process yield 10 samples for every sample. The dataset was augmented (Shorten and Khoshgoftaar, 2019) by repeating the whole process after (a) left–right flipping, (b) up–down flipping, (c) rotating 90°, (d) rotating 90° plus left–right flipping and (e) rotating 90° plus up–down flipping the full sample. At the end of the whole process we had 60 images of size 200×200 for every labeled 300×300 pixel sample. Since we labeled 239, we generated 14340 images.

The resulting dataset was slightly skewed. Some of the labeled samples came from canvases with similar fabrics, hence increasing the number of samples of the same type. To prevent the model to overfit to them, we generated twice the samples for the other paintings, including them in the dataset. By repeating the generation of samples for these fabrics we increased the number of images to 21540.

4. DL model

We designed a DL model to detect and segment crossing points between vertical and horizontal threads. The U-Net architecture (Ronneberger et al., 2015; Ali et al., 2022) was the starting point to build the model, see Fig. 3, where a top-down path, usually denoted as encoder or contracting path, is followed by a set of down-top layers, the decoder or expansive path. Layers are composed of one main unit, i.e., a module with 2D convolutions, batch normalization and a ReLU activation function. We developed several models based on this architecture. In the following, we first describe the best model found, i.e., the one providing the best threads densities estimations for the validation subset. Then we outline other evaluated models.

4.1. Inception module

We first tried a fixed size 2D convolutional kernel in every layer, while the size could change from one layer to the next. However, the outcome of this model was improved by introducing inception. Our conjecture is that the variety of threads densities of the canvases, in the range 6 to 23 thr/cm, makes it hard for the model to locate the crossing points if one fixed-size kernel is used. On the contrary, by means of the inception paradigm, in the same layer we have convolutional kernels of several sizes. We used $k \times k$ kernels with k=3, 5 and 7, see Fig. 4. Note that similar ideas were adopted in Zhang et al. (2022), Ali et al. (2022) and Yamanakkanavar and Lee (2022). The inception module, see horizontal (blue) arrows within the encoder and decoder in Fig. 3, is a sequence of blocks where the first one is a convolution with $n_i \times 3$ different kernels, where n_i denotes the number of filters per layer and i the layer level from top to bottom, see Fig. 4. The results are concatenated at the output. Then a batch normalization is performed, followed by a ReLU activation function. We used $n_0 = 8$ filters for the first upper layer and scaled this number by 2 as we went deeper in the encoder, $n_{i+1} = 2n_i$. All convolutions were performed with the 'same' option, i.e., zero padding of size $\lfloor k/2 \rfloor$ was introduced around every input image to get the same size at the output after convolution.
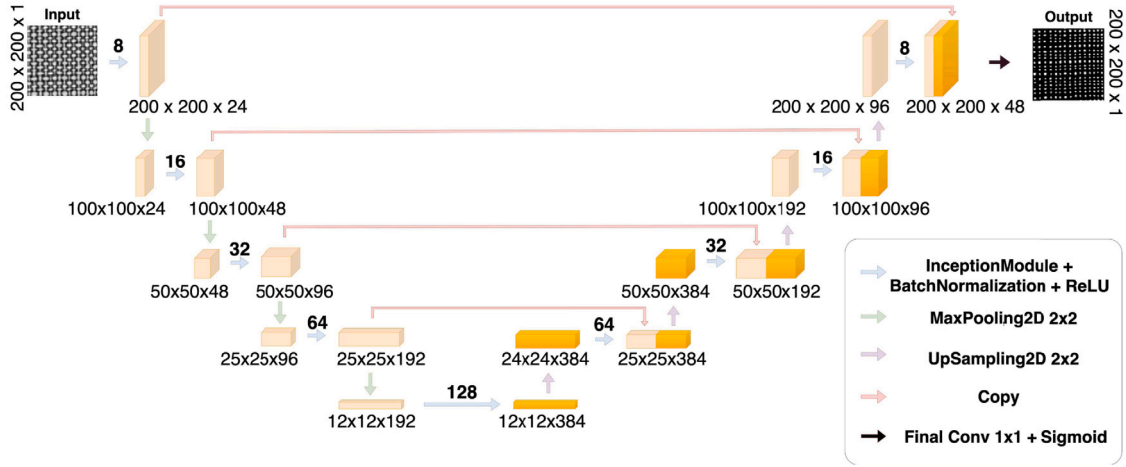
Fig. 3. The U-Net architecture used with the inception modules in blue horizontal arrows, with the number of filters used, n_i, indicated on top. Up and down steps involve halving reductions of sizes downwards, and double them upwards, with zero padding if needed. Skip (copy) connections concatenate the upcoming features with the features at the same level in the encoder. The size of all feature maps is indicated below them. At the output a 1×1 convolutional filter is used, with sigmoid activation function.
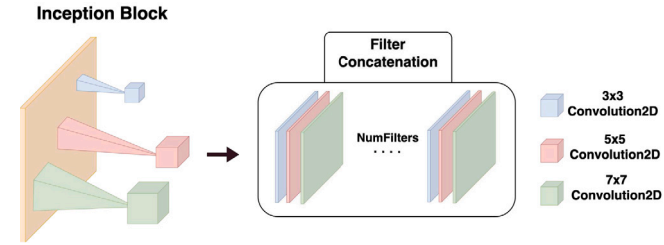

Fig. 4. Inception block: 3×3, 5×5 and 7×7 kernels are convolved with the input at the same depth, as many times as given by the number of filters parameter, n_i. The results of these convolutions are concatenated.

4.2. Model description

The model used, in Fig. 3, has an image of $200 \times 200 \times 1$ at the input layer. Then, in the encoder stage, the input undergoes 5 layers down to get a tensor of dimensions $12 \times 12 \times 384$. Each layer is composed of the following blocks:

1. Inception: three parallel 2D convolutional layers with square kernels of size 3, 5 and 7. Outputs of the inception are concatenated at the output, the number of features at the output of the inception is 3 times the number on top of the horizontal arrows in the encoder, that indicates the number of inception kernels.
2. Batch normalization.
3. ReLU activation function.
4. Max-pooling of stride 2 and kernel size 2, except for the last (5th) layer of the encoder.
5. Dropout of probability $p = 0.1$.

After the encoder, the resulting features go up through 4 layers in the decoder stage. Each layer in the decoder includes the following:

1. Upsampling of size 2×2, except for the last layer in the decoder (9th layer in the full network).
2. Inception: three parallel 2D convolutional layers with square kernels of size 3, 5 and 7. Outputs of the inception are concatenated at the output, the number of features at the output of the inception is 3 times the number on top of the horizontal arrows in the decoder, that indicates the number of inception kernels.
3. Batch normalization.
4. ReLU activation function.

5. Copy and concatenate the tensor at the output of the layer at the same level (same size) at the encoder, see long horizontal arrows joining the two vertical paths of the "U" shape in Fig. 3.
6. Dropout of probability $p = 0.1$.

Finally, at the output layer we have a 2D convolutional layer of kernel size 1 and a sigmoid as activation function. The outputs of the first 5 layers in the encoder have sizes $100 \times 100 \times 48$, $50 \times 50 \times 96$, $25 \times 25 \times 192$, $12 \times 12 \times 192$ and $12 \times 12 \times 384$, that are, in reverse order, the sizes of the inputs to the layers at the same level in the decoder and the output layer. The full model has over 6 million parameters.

4.3. Loss function

The binary cross-entropy was first used as loss function. However, when analyzing an image the output values of the model were far from being binarized and a thresholding was needed, where the Otsu (Otsu, 1979) method was applied. To provide an output with more extreme values we better used the Dice loss, then in the analysis stage the image was binarized with just a 0.5 threshold. Given the labeled output $\mathbf{I}$, a matrix of bits, and the estimated one $\hat{\mathbf{I}}$, the Dice loss is given by:

$$L_D = 1 - \frac{2 \sum_{i,j} \mathbf{I}[i,j]\hat{\mathbf{I}}[i,j]}{\sum_{i,j} \mathbf{I}[i,j] + \sum_{i,j} \hat{\mathbf{I}}[i,j]} \tag{1}$$

In Fig. 5 we included the output for a segmentation after training with binary cross-entropy and Dice loss. Note that the resulting output with Dice loss is a quasi-binary image, as needed later to estimate densities and angles of threads. We also observed better results in accuracy, used as validation loss, see Section 6. Therefore, in the learning stage of the model presented above, later used in the analysis of some case studies, we used the Dice loss in (1) as error to train the parameters of the networks and the accuracy as validation loss.

4.4. Other models

We described above the model that provided the best results in validation and the one used in the studies later included. Other three models were also proposed. The four of them share the number of layers, the max-pooling at every layer in the encoder, the sigmoid as activation layer in the output of the model and the ReLU as activation function in the other layers. The accuracy was used in all cases to validate the model. The main features of these other models are described in this subsection.
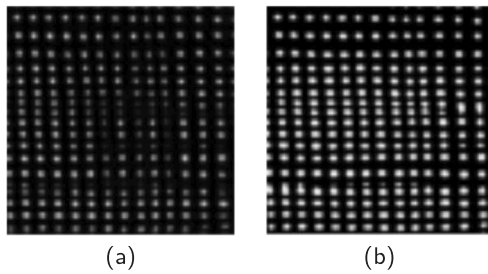
Fig. 5. Segmented image obtained with (a) binary cross-entropy as loss function and (b) Dice loss.

Fig. 6. Sketch of the search for $m = 9$ neighbors for a given crossing point (circled). The arrows to the neighbors indicate the vectors used to compute the angle deviations and the distances.

4.4.1. U-Net with Otsu

The U-Net model with layers of one kernel size was the first tried model. At the output, a threshold computed using the Otsu approach (Otsu, 1979) was applied prior to performing the thread densities estimations. In the encoder, the layer unit was a double repetition of convolutional plus batch normalization and ReLU blocks. All kernels size where $k = 3$ except for the first layer, set to $k = 7$. The initial number of filter was $n = 14$ and this number was doubled in the next layer, $n_{i+1} = 2n_i$. The dropout was set to $p = 0.25$ and the learning rate to 10^{-4}. These values were the ones exhibiting the best results. In the decoder, the transposed convolution was applied to upsample the intermediate results. The loss function used was the binary cross-entropy.

4.4.2. U-Net with Dice

This model was equivalent to the previous one but for the loss function, where the Dice in (1) was used instead. As the result was already quite binarized a 0.5 threshold was used at the output, to estimate the densities. The learning rate was increased to 10^{-3}.

4.4.3. Original inception

Another model used was a closer version of the inception model in Szegedy et al. (2015). As inception module we included four parallel convolutional layers:

1. A $k = 1$ convolution with batch normalization and ReLU.
2. A $k = 1$ convolution with batch normalization and ReLU followed by a $k = 3$ convolution with batch normalization and ReLU.
3. A $k = 1$ convolution with batch normalization and ReLU followed by a $k = 3$ convolution, another $k = 3$ convolution, batch normalization and ReLU.
4. A max-pooling of size 3×3 but with stride one, followed by a $k = 1$ convolution with batch normalization and ReLU.

The output of these four sequences of blocks were concatenated at the output. The learning rate was set to 10^{-3}.

4.5. SW and HW details

The models described above were programmed using Python 3.9 and Keras-Tensorflow 2.5.0. The input grayscale image size was $200 \times 200 \times 1$ and the default batch size was 32. We used the Adam optimizer (Kingma and Ba, 2015) with a learning rate value of 10^{-3}. An early stopping was also included with a latency of 5. A Tesla P100 with 16 GB memory was used along with a Intel Xeon E5-2630 v4 with 20 cores CPU.
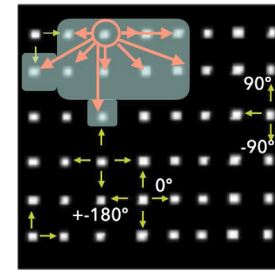
5. Density and angle estimations

At the output of our model we have an image with high values at the crossing points areas, see Fig. 5.(b) for an example, where a 1 cm side image is included. After binarization we locate these areas by computing their centroids. Next, we propose to estimate the densities of the vertical and horizontal threads as described in Algorithm 1, where $\bar{\mathbf{h}}$ denotes the mean value of vector $\mathbf{h}$ and we exploited the fact that inputs images have a resolution of 200 pixels per cm. In this work we used $m = 9$, $\alpha = 25°$ and $q = 10$ or $q = 0$. For every crossing point found, the method searches for the m nearby segmented crossing points to estimate distances and angles. In Fig. 6 we include an example where for a crossing point, circled in red, the $m = 9$ nearest ones are found. Within these m neighbors, the nearest ones above ($+90°$), below ($-90°$), to the right ($0°$) and to the left ($180°$) are found, if any. See Fig. 6 where some of these points have been marked with arrows. The averaged distances to the neighbors for all the crossing points are used to estimate the thread densities. Before averaging, removing values below and above a percentile, given by parameter q, might eliminate possible outliers. Angle orientations of the vertical and horizontal threads can also be estimated by checking the angles of the vectors pointing to the nearest neighbors.

Another approach to estimate the densities could be applying frequency analysis, i.e., the FT, to the result of the segmentation. We will refer to this method as FA.

6. Training and testing

6.1. Training models

We trained the proposed models ten times with different random initialization of the parameters, i.e., the kernels, for (A) the inception with Dice loss function (Inc-Dice) in Section 4.1, (B) the U-net with Otsu threshold (Unet-Th) in Section 4.4.1, (C) the U-Net with Dice (Unet-Dice) in Section 4.4.2 and (D) the original inception with Dice in Section 4.4.3 (Orig-Inc-Dice). In Fig. 7 we include a box diagram with the accuracy results of the segmentation for the validation set after training, with $q = 10$. The diamonds indicate outliers. It can be observed that the inception with Dice loss provides the best values, in mean and Q1 and Q3 quartiles. The Unet-Th does a good job, although with a higher dispersion. This dispersion is reduced with the Unet-Dice. The original inception exhibits a much worse result in most of the trainings, being the most computationally demanding, in view of the description in Section 4.4.3.

In Fig. 8 the box diagram is represented for the error in the counting using the SC approach with $q = 0$ for the trained models and validation set. The values for the horizontal and vertical densities were compared to the ones after applying the same SC approach to the ground truth (annotated samples with crossing-points). We measured the mean of the absolute error normalized by the true value, to properly highlight errors in high densities. As expected, the median values increased in the

Algorithm 1 Spatial Counting (SC)

Input: the number of segmented crossing points, M, and the centroids of the crossing points.

    for k = 1 to M **do** estimate the m nearest centroids to the kth one and store the distances to them, in pixels, as the kth row of the $M \times m$ matrix **D**.

    end for

    for k = 1 to M **do** estimate the angles of the vectors joining the kth centroid to the m nearest ones and store the result as the kth row of the $M \times m$ matrix **A**.

    end for

    Initialize **h** and **v** as empty vectors.

    for k = 1 to M **do**

        Search for the entries of the kth row of **A** with values in the range $[90 - \alpha, 90 + \alpha]$ and, if any, store in **v** the lowest distance found within the corresponding entries of the kth row of **D**.

        Search for the entries of the kth row of **A** with values in the range $[-90 - \alpha, -90 + \alpha]$ and, if any, store in **v** the lowest distance found within the corresponding entries of the kth row of **D**.

        Search for the entries of the kth row of **A** with values in the range $[-\alpha, +\alpha]$ and, if any, store in **h** the lowest distance found within the corresponding entries of the kth row of **D**.

        Search for the entries of the kth row of **A** with values in the range $[180 - \alpha, 180 + \alpha]$ and, if any, store in **h** the lowest distance found within the corresponding entries of the kth row of **D**.

    end for

    Remove from **h** and **v** values below the qth percentile and above the $(100 - q)$th percentile.

    Compute the horizontal, h, and vertical, v, thread densities, in threads/cm, as

$$h = 200/\overline{\mathbf{h}} \qquad (2)$$

$$v = 200/\overline{\mathbf{v}} \qquad (3)$$

    Likewise, the average tilt of the vertical and horizontal threads can be calculated by means of the average of all the angles of the corresponding crossing points in **v** and **h**, respectively.

---

same order as the validation decreased for the Inc-Dice, Unet-Th, Unet-Dice and Orig-Inc-Dice approaches. The training providing the lowest value was in the Inc-Dice set, with a 1.12% error. In the following we will use the Inc-Dice with these weights and SC, hereafter denoted by DLSC, to evaluate the performance in the test set and to analyze the case studies proposed.

The number of epochs varies from one training to the other. For the Inc-Dice the averaged number of epochs run was $8.7 + 5 \approx 14$ where 5 stands for the patience used in the early stopping. Since the training of the Inc-Dice lasted 180 s per epoch, each training of this model took approximately 42 minutes.

6.2. Test results

We evaluate the selected model, DLSC, for the test set. The averaged normalized absolute value for test was 1.83% with $q = 0\%$ and 1.61% if SC with $q = 10\%$ was used. This last value was the one selected for the processing of the whole X-ray of canvases. In Fig. 10 we include the error of this DLSC model for images in the test set. We show the absolute normalized error for the samples in the four 1×1 cm corners of the annotated 1.5×1.5 cm samples in the test set, both for the horizontal and vertical threads. In Fig. 9 we include the input (first column), the labeling (second column) and the output of the DL model (third column) results for images number 6 (Fig. 9.(a)–(c)) and 32 (Fig. 9.(d)–(f)) in Fig. 10, the ones with the largest errors in the horizontal and vertical densities, respectively. In the first case,
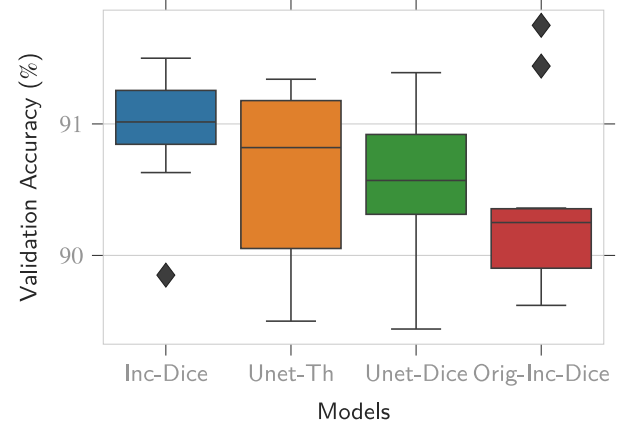

Fig. 7. Box diagram of accuracy results for the four models trained: Inc-Dice, Unet-Th, Unet-Dice and Orig-Inc-Dice.
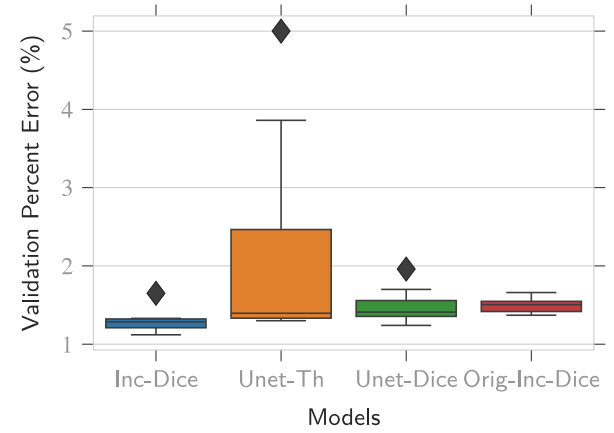

Fig. 8. Box diagram of normalized absolute errors in the density estimations for the four models trained: Inc-Dice, Unet-Th, Unet-Dice and Orig-Inc-Dice.
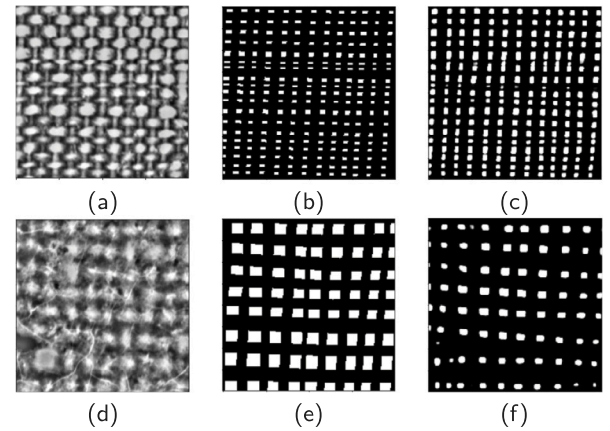

Fig. 9. The two images with worst errors in the test set in Fig. 10 for the horizontal (a) and vertical (d) densities. In (b) and (e) we have the annotations while in (c) and (f) the output of the DL segmentation.

the very thin treads are hard to segment, providing a value of 17.67 thr/cm when the true value was 19.18 thr/cm. In the second scenario, (I) the poor quality of the image and (II) the error in the labeling where some crossing points are missing cause this error, estimating
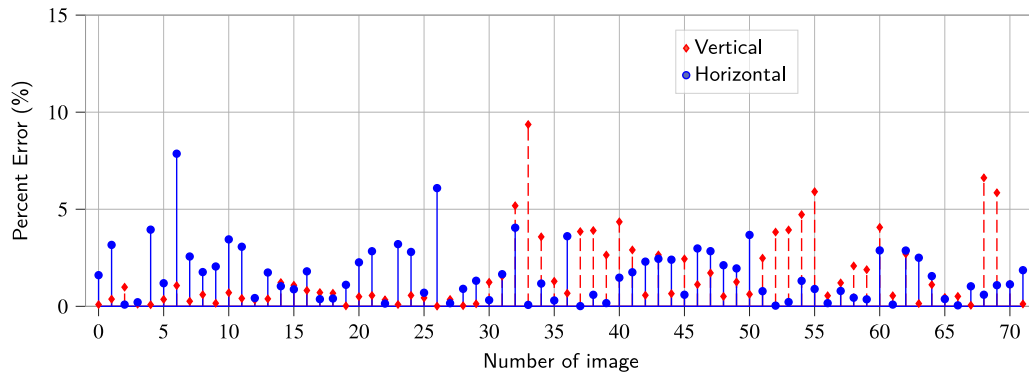
Fig. 10. Normalized absolute error (%) in horizontal (○ solid) and vertical (◇ dashed) densities for the 1 × 1 cm corners of the 1.5 × 1.5 cm annotated samples in the test set, DL and SC with $q = 10\%$ was used.
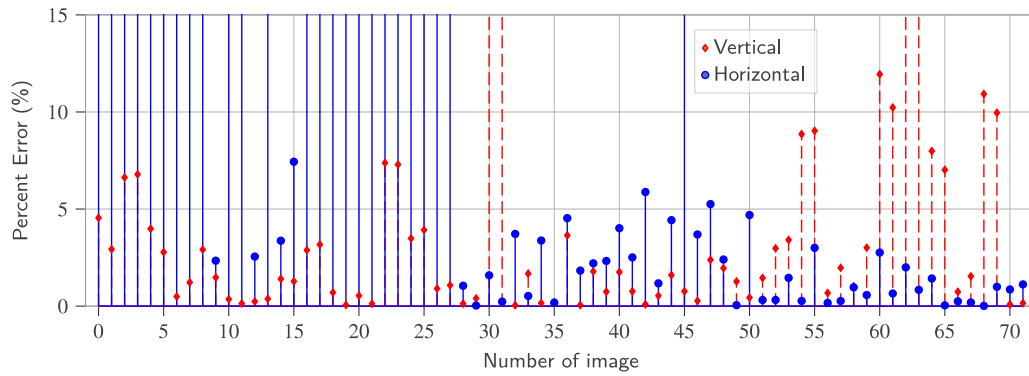

Fig. 11. Normalized absolute error (%) in horizontal (○ solid) and vertical (◇ dashed) densities for the 1 × 1 cm corners of the 1.5 × 1.5 cm annotated samples in the test set and the FT approach.

9.36 thr/cm from the segmentation where 8.9 thr/cm is estimated from the annotations. See the left lower and upper parts in Fig. 9.(d)–(f) as examples of (I) and (II), respectively.

In Fig. 11 we include the results for the FT approach applied to the same test images using 2048 points for the discrete FT. It can be observed how the results for the first 28 images are quite poor, normalized errors rise up to 50%, being more relevant for the horizontal threads, as vertical ones are better observed in the images. The overall average normalized error is 7.47%. These first 28 images came from the 'The Crucified Christ' by Velázquez (P001167 in MNP inventory) (de Silva y Velázquez, 1632a) where we have a similar aspect to the one in Fig. 9.(a), i.e., high thread densities and low noise, and the horizontal threads cannot be visually identified as lines. For the rest of the test images, to the right, both warp and weft are better observed. However, the FT still provides worse results. For example, above image number 28 in Fig. 11 we have eight images with error above 10% while in the DLSC, Fig. 9, we have none in the whole test set.

7. Canvas analysis

In the following, we propose two case studies, to check for the correspondence between pairs of fabrics by using the DLSC approach. We will also include the results of the FT method as described in Simois and Murillo-Fuentes (2018). We process 1 × 1 cm patches in the full preprocessed image, at 200 ppc, of the X-ray of the canvas, **Z**, whose top left corners are at $\mathbf{Z}[p \cdot s, q \cdot s]$ where p and q are non-negative integers and s is the shift, in pixels, from one patch to the following one.

7.1. Velázquez's portraits

We analyze two canvases by Velázquez, on the one hand Antonia de Ipeñarrieta y Galdós and her Son, Luis (P001196) (de Silva y Velázquez,

Fig. 12. Paintings by Velázquez (a) Antonia de Ipeñarrieta y Galdós and her Son, Luis (de Silva y Velázquez, 1632b) and (b) Diego del Corral y Arellano (de Silva y Velázquez, 1632c).

1632b) and on the other Diego del Corral y Arellano (P001195) (de Silva y Velázquez, 1632c). In this couple of canvases husband and wife were portrayed, see Fig. 12, and hence it is conjectured that both were painted at the same time on fabrics from the same roll.

We used the DLSC approach with $q = 10\%$. In Fig. 13 we include the horizontal 'density map' for both paintings. This density map is the estimated value for the threads counting along the canvas. With a shift

Table 1
Estimation of densities for the horizontal and vertical threads in thr/cm for the patches in Fig. 16 with DL based crossing points segmentation and SC (DLSC), DL based crossing points segmentation with FA (DLFA) and FT approach (FT).

Sample	Horizontal (thr/cm)			Vertical (thr/cm)		
	DLSC	DLFA	FT	DLSC	DLFA	FT
Fig. 16.(a)	16.88	16.83	11.74	19.50	17.19	12.89
Fig. 16.(c)	14.47	14.50	14.47	17.37	19.14	10.54

of $s = 100$, we have four pixels in the resulting density map for every square centimeter in the canvas. The color of any pixel encodes the value of the thread density in thr/cm in the corresponding location, $[ps, qs]$, see the color bar to the right. In the figure we observe a vertical dashed line. The image to the left corresponds to the horizontal density map of P001196, while the one to the right to P001195 horizontally flipped. It can be clearly observed how the pattern of variations of the separation in the horizontal threads perfectly matches in both canvases. This indicates that both fabrics came from the same roll.

In Fig. 14 the same analysis, performed with FT (Simois and Murillo-Fuentes, 2018), is included. It can be observed that the result is quite noisy and conclusions about fabric pairing are harder to draw. In Fig. 15.(a) and (b) we include the vertical density map for P001196 computed with FT and DLSC respectively. It can be observed that in the vertical density the noise if stronger when using FT, with a very high variability in the vertical axis, while in the DLSC case the result is quite more consistent with the fact that along any vertical line density should not significantly change. To illustrate the differences in the results of the new proposed approach and the FT we focus on two of the processed patches, in Fig. 16.(a) and (c). In Fig. 16.(b) and (d) we have the corresponding outputs of the DL segmentation model. The estimated densities values are analyzed in Table 1 for both samples and for the horizontal and vertical threads. Note that since these samples are of size 1×1 cm, the visual counting of horizontal and vertical threads provides the densities in thr/cm. The densities estimated by FT are quite poor in the case of the first sample, in Fig. 16.(a). The main reason, as pointed in Section 1, is that we have threads with different widths and that in this case vertical threads are poorly observed compared to the horizontal ones. Put in other words, we have both effects described through Fig. 1. This is the reason why, in this scenario, the main frequency found by the FT approach does not correspond to the thread counting. In Table 1 we have an estimation of 16.88 and 19.50 thr/cm for horizontal and vertical threads, respectively, by means of the DLSC approach, while the FT is providing 11.74 and 12.89 thr/cm, i.e., it is focusing more on the thick threads than on the thin ones. By using the DLSC, we avoid focusing on main frequencies to measure distances between any pair of consecutive threads. When the segmentation fails, for example because of the poor quality of the image, the estimation of densities through the average distances between threads is not accurate. In Fig. 16.(c) the fabric cannot be observed in a large part of the patch. In the output of the segmentation, Fig. 16.(d), we observe that the crossing points are missing in some parts. In the lower part of the output, the estimation of the vertical thread density by using SC introduces an error in the overall estimation, as observed in Table 1. Still, the DLSC is providing a much better estimation than the FT. However, if in this case the FA is used instead of the SC, we observe an improvement, estimating 19.14 thr/cm compared to 17.37 thr/cm of the DLSC and 10.54 thr/cm of the FT. Usually, we observe outcomes of the DLFA better than the FT but worse than the DLSC while at some points, as in this patch, the DLFA provides better estimations.

Finally, in Fig. 17.(a) and (b) we include the angle deviation estimations for P001195 of horizontal and vertical threads, respectively. It can be observed the 'garland' effect, i.e., the periodic variation of deviations near the sides of the canvas, due to the separation between nails. This indicates that the painting is conserved in its original size. Also, in the left upper corner of Fig. 17.(b) it can be observed that
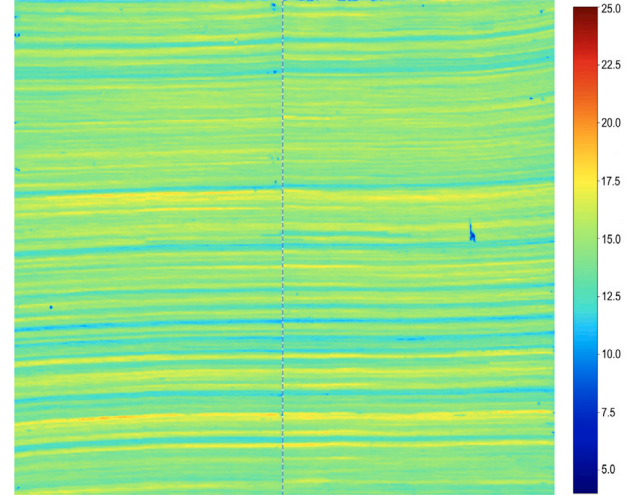

Fig. 13. Match of horizontal thread densities using DLSC, in thr/cm, for Antonia de Ipeñarrieta and Son (P001196), to the left and Diego del Corral *y* Arellano (P001195), flipped horizontally, to the right. The maps of densities are separated by a dashed line.
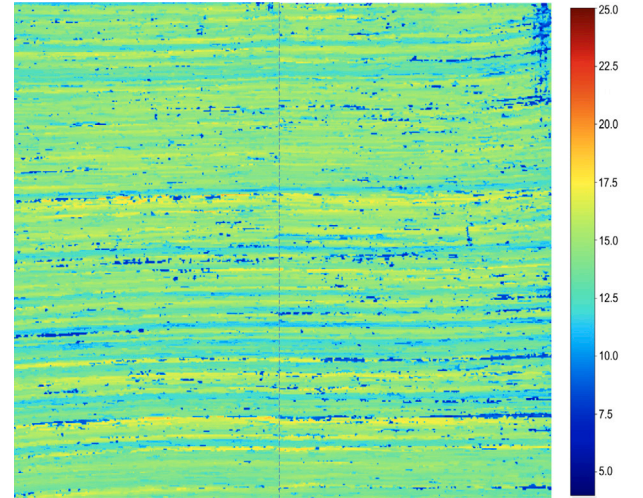

Fig. 14. Match of horizontal thread densities using FT, in thr/cm, for Antonia de Ipeñarrieta and Son (P001196), to the left and Diego del Corral *y* Arellano (P001195), flipped horizontally, to the right. The maps of densities are separated by a dashed line.

the fabric is twisted. This is consistent with the deformation of lines of thread densities in Fig. 13, right upper corner.

The inference in this case took approximately 43 min for each canvas, in the GPU Tesla P100 - CPU Intel Xeon HW. The time to perform inference depends on the number of patches to be processed. As already mentioned we used a 50% overlap, since the canvas is 2.12×1.21 m size, we run the Inc-Dice model plus the spatial counting approach for $212 \times 121 \times 4 = 102608$ patches.
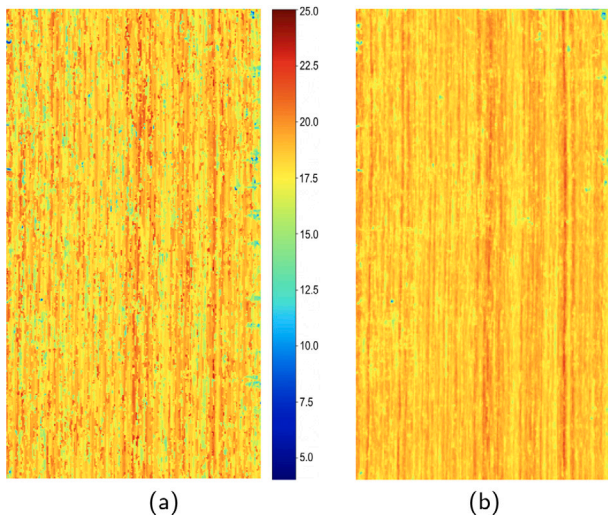
Fig. 15. Vertical thread densities, in thr/cm, for Antonia de Ipeñarrieta and Son (P001196) with (a) FT and (b) DLSC.
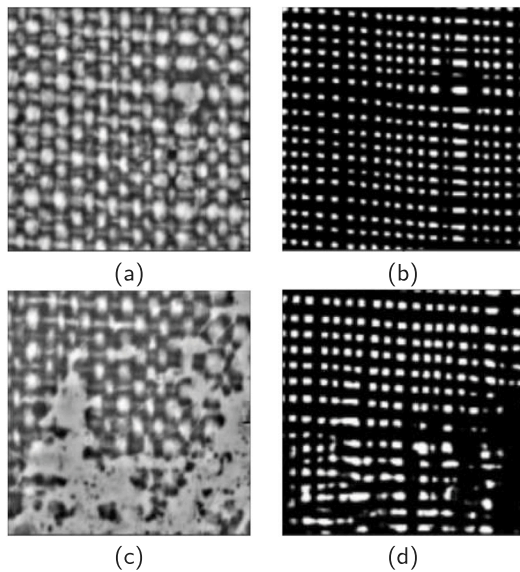

Fig. 16. In (a) and (c) patches of 1×1 cm of the X-ray plate of Diego del Corral *y* Arellano (P0011195) and in (b) and (d) the corresponding outputs of the DL based segmentation.

7.2. Murillo's prodigal son

In the studies of series of canvases the thread density analysis proves to be useful to check for the use of the same fabric in the paintings. We bring here two masterpieces produced by Murillo for the series illustrating the biblical parable of the prodigal son: "The prodigal son taking leave of his home" (P00998) (Murillo, 1660-1965a) and "The prodigal son squandering his inheritance" (P00999) (Murillo, 1660-1965b), see Fig. 18. In Fig. 18 we also include to the right the results of the vertical thread density estimation with $s = 20$ using DLSC, where P00999 has been rotated 180 degrees. It can be observed the good match between the fabrics. In Fig. 19 results for the FT and DLSC applied to the canvas P00999 can be compared both for the horizontal (top) and vertical (bottom) thread densities. It is interesting to note the improved definition of the DLSC approach, better allowing for the identification of color lines, representing densities along the warp and weft.
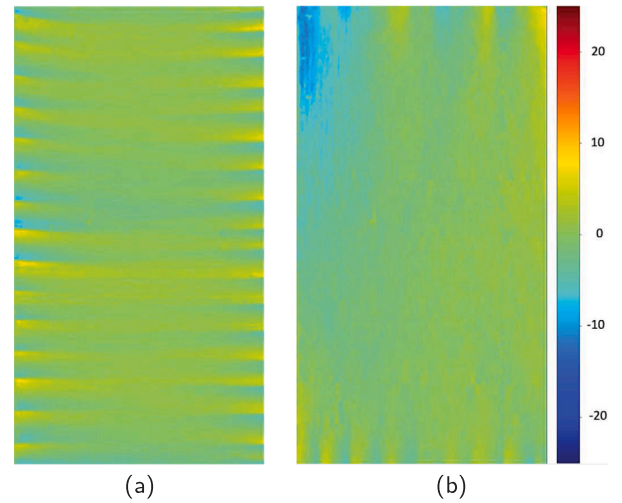

Fig. 17. Angle deviation, in degrees, for Diego del Corral *y* Arellano, of (a) horizontal and (b) vertical threads.

7.3. High intensity and low contrast areas

In Fig. 20 we include the results for the estimation of the densities of horizontal threads for one of the X-ray plates of the canvas Ixion by Ribera (1632). The processed area within the whole canvas is highlighted in Fig. 20.(a) while a 1 cm side patch is included in Fig. 20.(b). The results for the FT and DLSC are reported in Fig. 20.(c) and Fig. 20.(d), respectively. It can be observed that the result of the DLSC is not as good as the one of the FT. Note that the fabric, see Fig. 20.(b), has a low density of threads with a very regular distance between them. In this scenario the FT is quite robust and provides excellent results. On the contrary, the DLSC fails in areas of the plates where we have very high levels and low contrast, because threads cannot be observed in the image. Please pay attention to the shoulder of the man on top and the high intensity area below its arm, within the processed area. This degrades the outcome of the segmentation, and the SC is unable to provide an accurate result as it needs a good enough grid of crossing points. However, a method searching for main frequencies after the segmentation succeeds, see the outcome of the DLFA in Fig. 20.(e). It can be observed that we achieve similar results to the ones in Fig. 20.(c), with the FT. But in these areas, the DL will not improve the FT.

8. Conclusions

In this paper we present a multidisciplinary investigation in which an effort has been made to understand the problem posed, analyze the state of the art, its drawbacks and study the possible alternatives. In this process we identify and report two scenarios in which methods based on frequency domain analysis fail. To overcome these problems, we propose to resort to the spatial domain by segmenting the crossing points.

The previous known solution working in the spatial domain needs a previous partial labeling of the image to be analyzed. We propose a new algorithm that can use the curator with minimal effort and knowledge, without labeling or a complex parameter selection.

The U-Net, used as starting point, was not capable of processing fabrics with different thread densities, typically in the 5-25 range. Fixed width kernels did not give good results in all cases. We investigated several models with different layers and kernel sizes to conclude that inception was a good option to accommodate the different thread frequencies possible. We also found that the loss functions used in training had a relevant impact on the result. Consequently, we studied
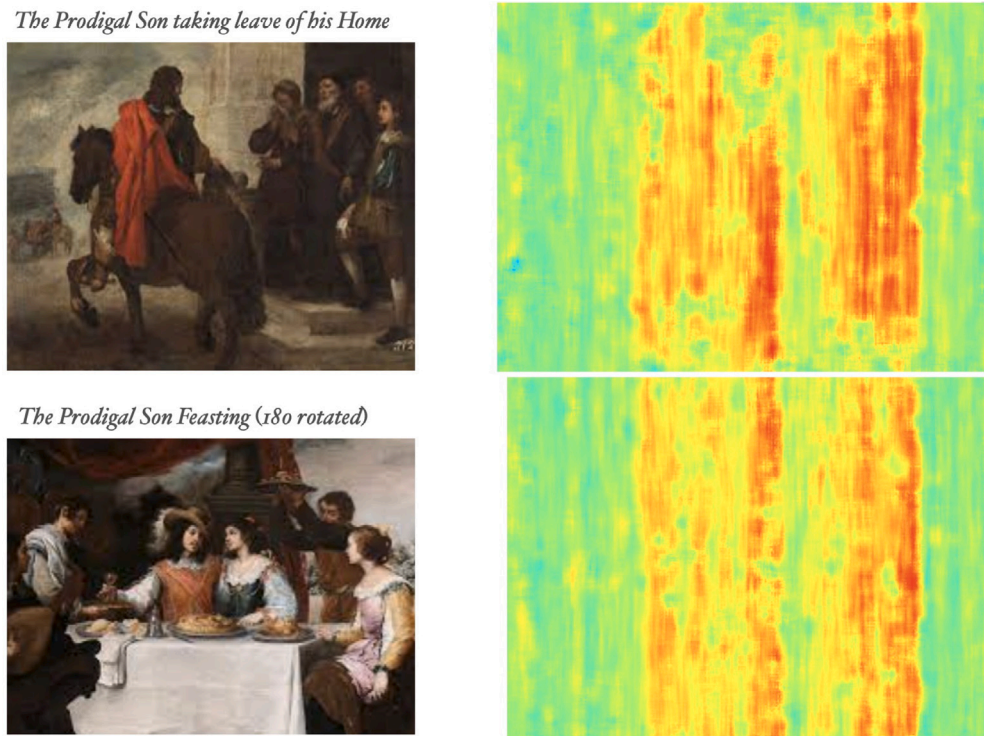
Fig. 18. Canvases (left) and vertical thread densities (right), in thr/cm, for "The prodigal son taking leave of his home" P00998 (top) and "The prodigal son squandering his inheritance" P00999 (bottom, rotated 180°).
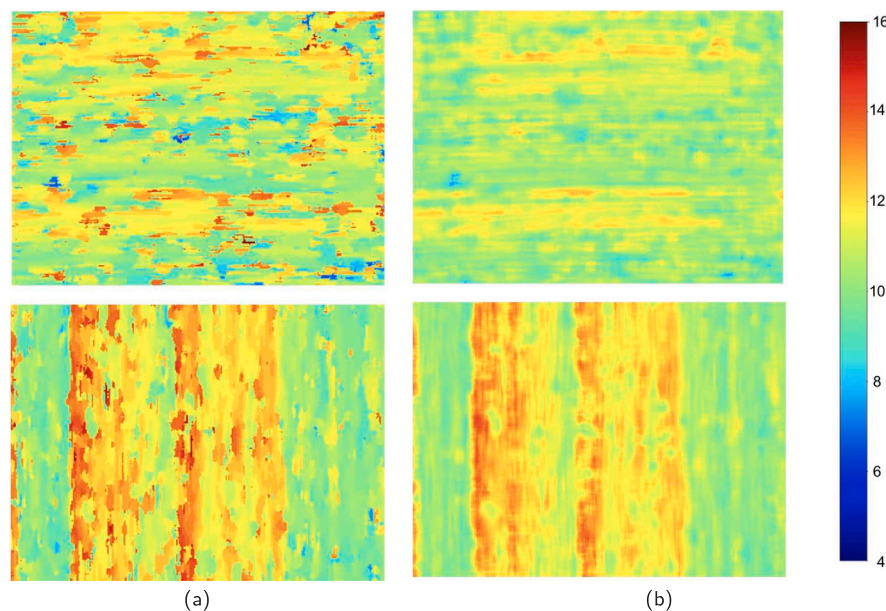

Fig. 19. Horizontal (top) and vertical (bottom) thread densities, in thr/cm, for P00999 with (a) FT and (b) DLSC.

the different possibilities and adopted Dice as the criterion to evaluate the error of the model.

The generation of the dataset from scratch, another contribution of this work, is also developed. This includes the preprocessing and the data augmentation stages. Labeling of the data is cumbersome. To gain in efficiency, we decided to label larger samples (1.5 times wider and taller) than needed at the input of the model, to then use different areas in the data augmentation step. The organization of the dataset into training, validation and test was another problem to be addressed. At the beginning of our research we included samples of the same painting in the all of them. Although this has the advantage of having fairly representative subsets, it does not guarantee good performance when a new canvas is presented. It must be taken into account that canvases from different authors may present different counts, different qualities of fabrics or different primers used. We designed a partition to ensure that (1) canvases of different thread densities are included in the subsets and (2) different qualities are considered in the training, validation and testing. In addition, since a small number of instances with very high and very low thread densities are available, in the training phase we had to balance the number of samples used from each image to avoid skewness.
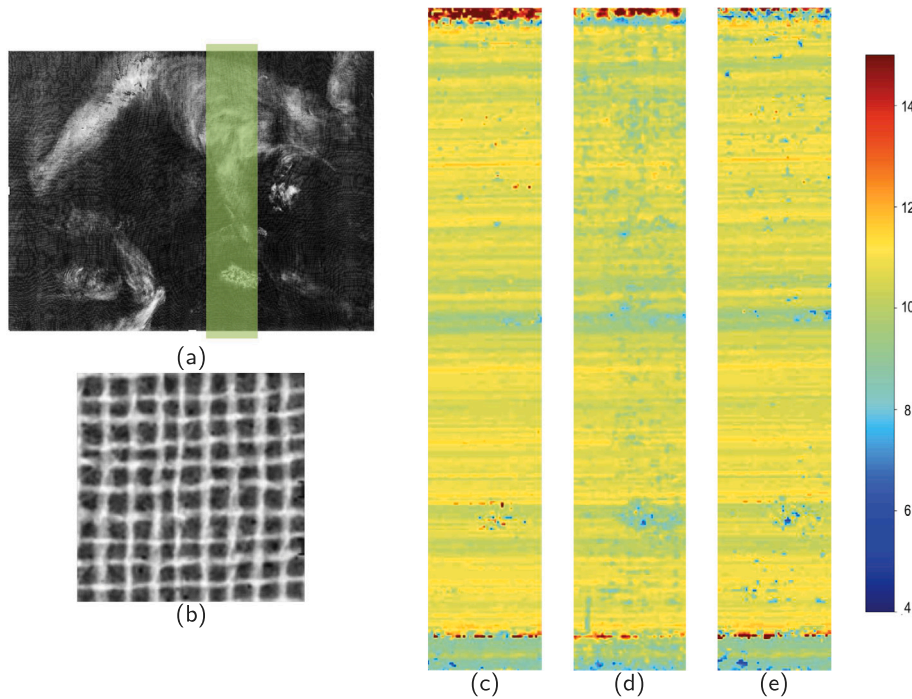
Fig. 20. Horizontal thread densities, in thr/cm, for the vertical area highlighted in (a) over the full Xray of Ixion by Ribera (1632) with (c) FT, (d) DLSC and (e) DLFA. A 1 cm side patch is included in (b).

The proposed deep learning model, after training, provides the crossing points found in the fabric, but we needed to translate the result into the horizontal and vertical counts. We presented an approach to cope with this, the SC, that also provides the estimation of the angle deviations of the threads. Then we checked the performance at the output of the SC, to observe how a result in segmentation translated into thread densities estimations. Our model selection relied on the final count performance, as discussed in Section 6.1. In Section 6.2 we carefully analyzed the final outcome for the patches in the test subset, coming from three new canvases of different densities. In the test set, the whole DLSC procedure, segmentation plus spatial counting, exhibited a very low normalized absolute error, 1.61%, while the FT approach had 7.47%.

To illustrate the good performance of the novel approach, we present two case studies. On the one hand, we include a detailed analysis of the canvases P01195 (de Silva y Velázquez, 1632c) and P01196 (de Silva y Velázquez, 1632b) by Velázquez. In this comparison of densities, the FT fails to produce a fine enough density estimation maps, as predicted, while the proposed approach provides a very useful result. We also included a comparison of a pair of canvases of the series The Prodigal Son, by Murillo, to prove that both canvases were painted on canvases coming from the same roll. Again, our new approach exhibited neater results for the density maps, allowing for a better comparison.

While the DLSC proposed approach presented quite good results in some scenarios, in others its outcome was worse compared to the FT method. It failed for some patches of P001195 by Velázquez or some areas of P001114 by Ribera where the fabric could hardly been observed. In these cases, the DLFA, where frequency analysis was used instead of the SC method, could be used. We conjecture that with a better preprocessing stage at the segmentation input we could further improve the DLSC approach. This is a future line of research.

To sum up, this novel proposal focuses on the application of deep learning to thread counting in old paintings. This works starts with a careful problem analysis to later face a full design from the dataset design to the spatial thread counting, paying attention to the deep model development and the loss functions used. As a result, we have

a method that improves the state of the art FT approach in several relevant scenarios, with no need of pre-labeling.

CRediT authorship contribution statement

A. Delgado: Conceptualization, Investigation, Software, Data curation, Writing – review & editing. **Laura Alba-Carcelén:** Resources, Validation. **Juan J. Murillo-Fuentes:** Conceptualization, Investigation, Methodology, Software, Data curation, Writing – original draft, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Juan J. Murillo-Fuentes reports financial support was provided by Junta de Andalucía - European Union (FEDER).

Data availability

The authors do not have permission to share data.

Appendix. Preprocessing

The images were enhanced with algorithms based on their local mean and variance when the patches were cropped from the plates to facilitate the labeling and the training of the network. The preprocessing was done in three stages:

1. *Local mean filtering.* In the local mean filtering we first compute the average around a pixel then subtract this value to the value of the pixel. The computation of the average value is computed by convolving with a constant value square kernel of size $w = 2s + 1$. If $\mathbf{X}$ is the input image and $\mathbf{X}[i, j]$ is the pixel in the ith row and jth column, the result of this step is

$$\mathbf{Y}[i, j] = \mathbf{X}[i, j] - \frac{1}{w^2} \sum_{k=i-s}^{i+s} \sum_{l=j-s}^{j+s} \mathbf{X}[k, l] \tag{4}$$

With the local mean filtering we avoid changes in intensities due to wood stretches or to areas with more opaque paintings. In this work we used $w = 13$.

2. *Standard deviation filtering.* Then, to ensure that the whole range is used we divide by the local standard deviation, avoiding division by zero. We first compute the variance as

$$\sigma_{i,j}^2 = \frac{1}{w^2} \sum_{k=i-s}^{i+s} \sum_{l=j-s}^{j+s} (\mathbf{Y}[k,l])^2 \tag{5}$$

where we also used $w = 13$. Then the output of this stage is

$$\mathbf{Z}[i,j] = \mathbf{Y}[i,j] / \max(\sigma_{i,j}, \epsilon) \tag{6}$$

where ϵ is any tiny value.

3. *Clipping and Scaling.* In this step we scale the image, $\mathbf{Z}$, to use the full dynamic range by setting its lowest and largest values to zero and one, respectively. Prior to this step we clip low probable largest and lowest values as follows. We first estimate the probabilities of values to be into one out of 256 segments of same lengths between the minimum and maximum values. Extreme values with probabilities below some threshold, γ, are clipped. We used $\gamma = 10^{-3}$.

References

Alba, L., Murillo-Fuentes, J.J., 2021. Fabrics as a painting support. new tools for the study. In: La Ciencia Y El Arte. Ciencias Experimentales Y Conservación Del Patrimonio. Ministerio de Cultura y Deporte, pp. 219–230. http://dx.doi.org/10.1007/978-3-319-75316-4_7.

Ali, R., Chuah, J.H., Talip, M.S.A., Mokhtar, N., Shoaib, M.A., 2022. Crack segmentation network using additive attention gate—CSN-II. Eng. Appl. Artif. Intell. 114, 105130.

Aradillas, J.C., Murillo-Fuentes, J.J., Olmos, P.M., 2021. Boosting offline handwritten text recognition in historical documents with few labeled lines. IEEE Access 9, 76674–76688.

Barni, M., Pelagotti, A., Piva, A., 2005. Image processing for the analysis and conservation of paintings: Opportunities and challenges. IEEE Signal Process. Mag..

Cornelis, B., Yang, H., Goodfriend, A., Ocon, N., Lu, J., Daubechies, I., 2017. Removal of canvas patterns in digital acquisitions of paintings. IEEE Trans. Image Process. 26, 160–171.

de Carbonnel, K.V., 1980. A study of french painting canvases. J. Am. Inst. Conserv. 20, 3–20.

de Silva y Velázquez, D.R., 1632a. The crucified Christ. Museo Nacional del Prado (P001167).

de Silva y Velázquez, D.R., 1632b. Antonia de ipeñarrieta y Galdós and her son. Luis, Museo Nacional del Prado (P001196).

de Silva y Velázquez, D.R., 1632c. Diego del Corral y Arellano. Museo Nacional del Prado (P001195).

de Silva y Velázquez, D.R., 1634. Prince Baltasar Carlos on horseback, Museo Nacional del Prado (P001180).

Deligiannis, N., Mota, J.F.C., Cornelis, B., Rodrigues, M.R.D., Daubechies, I., 2017. Multi-modal dictionary learning for image separation with application in art investigation. IEEE Trans. Image Process. 26, 751–764.

Escofet, J., Millán, M.S., Ralló, M., 2001. Modeling of woven fabric structures based on Fourier image analysis. Appl. Opt..

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.

Johnson, D.H., Erdmann, Jr., C.R.J.R.G., 2013. Weave analysis of paintings on canvas from radiographs. Signal Process. 93, 527–540.

Johnson, C., Hendriks, E., Berezhnoy, I., Brevdo, E., Hughes, S., Daubechies, I., Li, J., Postma, E., Wang, J., 2008. Image processing for artist identification. IEEE Signal Process. Mag..

Johnson, D.H., Sun, L., Johnson, C.R., Hendriks, E., 2010. Matching canvas weave patterns from processing X-ray images of master paintings. In: IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, http://dx.doi.org/10.1109/ICASSP.2010.5495297.

Kingma, D.P., Ba, J., 2015. Adam: A Method for Stochastic Optimization. In: Conf. for Learning Representations, San Diego, California, USA. URL http://arxiv.org/abs/1412.6980.

Lee, J.-S., 1980. Digital image enhancement and noise filtering by use of local statistics. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-2 165–168.

Maaten, L., Erdmann, R.G., 2015. Automatic thread-level canvas analysis: A machine-learning approach to analyzing the canvas of paintings. IEEE Signal Process. Mag..

Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., Terzopoulos, D., 2022. Image segmentation using deep learning: A survey. IEEE Trans. Pattern Anal. Mach. Intell. 44, 3523–3542.

Murillo, B.E., 1660-1965a. The prodigal son taking leave of his home. Museo Nacional del Prado (P00998).

Murillo, B.E., 1660-1965b. The prodigal son squandering his inheritance. Museo Nacional del Prado (P00999).

Nemade, R., Nitsure, A., Hirve, P., Mane, S.B., 2017. Detection of forgery in art paintings using machine learning.

Otsu, N., 1979. A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. 9, 62–66.

Polatkan, G., Jafarpour, S., Brasoveanu, A., Hughes, S., Daubechies, I., 2009. Detection of forgery in paintings using supervised learning. In: 2009 16th IEEE International Conference on Image Processing. ICIP, pp. 2921–2924. http://dx.doi.org/10.1109/ICIP.2009.5413338.

Pu, W., Sober, B., Daly, N., Higgitt, C., Daubechies, I., Rodrigues, M.R.D., 2020. A connected auto-encoders based approach for image separation with side information: With applications to art investigation. In: IEEE Int. Conf. on Acoustics, Speech and Signal Process.. ICASSP, pp. 2213–2217.

Ribera, L.S., 1632. Jusepe de, Ixion. Museo Nacional del Prado (P01114).

Roberto, J., Ortego, D., Davis, B., 2020. Toward the automatic retrieval and annotation of outsider art images: A preliminary statement. In: AI4HI.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional networks for biomedical image segmentation. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). http://dx.doi.org/10.1007/978-3-319-24574-4_28.

Rubens, P.P., 1628. Andan, and Eva Museo Nacional del Prado (p001692).

Rucoba-Calderón, C., Ramos, E., Gutiérrez-Cárdenas, J., 2022. Crack detection in oil paintings using morphological filters and K-SVD algorithm. In: Lossio-Ventura, J.A., Valverde-Rebaza, J., Díaz, E., Muñante, D., Gavidia-Calderon, C., Valejo, A.D.B., Alatrista-Salas, H. (Eds.), Information Management and Big Data. Springer International Publishing, Cham, pp. 329–339.

Rumelhart, D., Hinton, G., Williams, R., 1986. Neurocomputing: foundations of research, learning internal representations by error propagation.

Shorten, C., Khoshgoftaar, T.M., 2019. A survey on image data augmentation for deep learning. J. Big Data 6, 1–48.

Simois, F.J., Murillo-Fuentes, J.J., 2018. On the power spectral density applied to the analysis of old canvases. Signal Process. 143, 253–268.

Sizyakin, R., Cornelis, B., Meeus, L., Dubois, H., Martens, M., Voronin, V., Pizurica, A., 2020. Crack detection in paintings using convolutional neural networks. IEEE Access 8, 74535–74552, Cited by: 8; All Open Access, Gold Open Access, Green Open Access.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 1–9. http://dx.doi.org/10.1109/CVPR.2015.7298594.

Yamanakkanavar, N., Lee, B., 2022. Mf2-net: A multipath feature fusion network for medical image segmentation. Eng. Appl. Artif. Intell. 114, 105004.

Yang, H., Lu, J., Brown, W.P., Daubechies, I., Ying, L., 2015. Quantitative canvas weave analysis using 2-D synchrosqueezed transforms: Application of time-frequency analysis to art investigation. IEEE Signal Process. Mag. 32, 55–63.

Zhang, D., Zhao, J., Chen, J., Zhou, Y., Shi, B., Yao, R., 2022. Edge-aware and spectral–spatial information aggregation network for multispectral image semantic segmentation. Eng. Appl. Artif. Intell. 114, 105070.

Zou, Z., Zhao, P., Zhao, X., 2021. Virtual restoration of the colored paintings on weathered beams in the forbidden city using multiple deep learning algorithms. Adv. Eng. Inform. 50, Cited by: 0.