به نام خدا

نام: فاطمه زهرا بخشنده **استاد: دکتر محمدرضا محمدی**

شماره دانشجویی: 98522157 **درس مبانی بینایی کامپیوتر**

گزارش تمرین 7:

سوال اول:

ابتدا 8 تصویر را خوانده و در یک ردیف نمایش می‌دهیم.

آبجکت stitcher را در OpenCV با مد PANORAMA می سازیم و سپس با فراخوانی تابع stitch آن، لیست تصاویر را به آن می دهیم. اگر status در خروجی 0 باشد یعنی stitching با موفقیت انجام شده است. در غیر این صورت احتمالاً به دلیل عدم شناسایی نقاط کلیدی کافی، عملیات fail شده است.

خروجی در صورت 0 بودن status، عکسی است که این 8 تصویر، با نقاط مشترکشان به هم متصل شده اند و این تصویر بزرگتر را تشکیل داده اند. (خاصیت موزاییک شدن)

منابع: لینک

سوال دوم:

این تبدیل دارای 1 پارامتر θ برای چرخش است، پس فقط به 1 نقطه و تبدیل آن نیاز داریم ولی ممکن است این نقطه و تبدیل آن، با خطا به یکدیگر متناظر شده باشند و باعث شود که تبدیل دچار خطا شود به همین دلیل از تمام نقاط کلیدی استفاده می‌کنیم تا تبدیل دقیق‌تری داشته باشیم. حال برای اندازه‌گیری خطا می توان از حداقل مربعات و بهینه سازی استفاده کرد.

حداقل مربعات:

$$cost = \sum (x_2^n - x_1^n \cos\theta + y_1^n \sin\theta)^2 + (y_2^n - x_1^n \sin\theta - y_1^n \cos\theta)^2$$

$$\frac{d}{d\theta} cost = 2 \sum (x_1^n \sin\theta + y_1^n \cos\theta)(x_2^n - x_1^n \cos\theta + y_1^n \sin\theta) + (-x_1^n \cos\theta + y_1^n \sin\theta)(y_2^n - x_1^n \sin\theta - y_1^n \cos\theta) = 0$$

$$=> \sum (x_2^n\, x_1^n \sin\theta - (x_1^n)^2 \sin\theta \cos\theta + x_1^n y_1^n (\sin\theta)^2 + x_2^n\, y_1^n \cos\theta - x_1^n y_1^n (\cos\theta)^2 + (y_1^n)^2 \sin\theta \cos\theta) + (-x_1^n\, y_2^n \cos\theta + (x_1^n)^2 \sin\theta \cos\theta + x_1^n y_1^n (\cos\theta)^2 + y_2^n\, y_1^n \sin\theta - x_1^n y_1^n (\sin\theta)^2 - (y_1^n)^2 \sin\theta \cos\theta) = 0$$

$$=> \sum \sin\theta\, (x_2^n\, x_1^n + y_2^n\, y_1^n) + \cos\theta\, (x_2^n\, y_1^n - x_1^n\, y_2^n) = 0$$

$$=> \sin\theta \sum (x_2^n\, x_1^n + y_2^n\, y_1^n) + \cos\theta \sum (x_2^n\, y_1^n - x_1^n\, y_2^n) = 0$$

$$=> \tan\theta = \frac{\sum (x_1^n\, y_2^n - x_2^n\, y_1^n)}{\sum (x_2^n\, x_1^n + y_2^n\, y_1^n)}$$

$$=> \theta = \tan^{-1} \frac{\sum (x_1^n\, y_2^n - x_2^n\, y_1^n)}{\sum (x_2^n\, x_1^n + y_2^n\, y_1^n)}$$

نکته: این روش به داده های پرت حساس است.

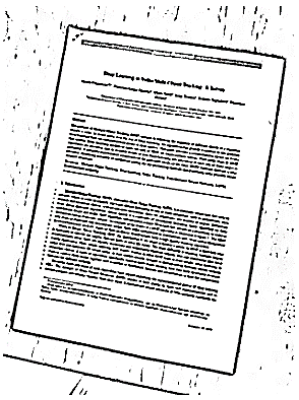
منابع: اسلاید

سوال سوم:

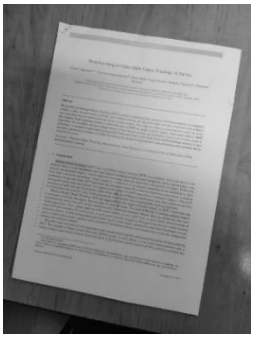
ابتدا تصویر را می‌خوانیم و نمایش می‌دهیم.

1- نگاشت سیاه و سفید: تصویر رنگی BGR را GrayScale کرده و در grayscale ذخیره می کنیم.

2- محو کردن تصویر: در این مرحله می‌خواهیم تصویر را برای مراحل بعدی به ویژه لبه یابی آماده کنیم. می توانیم از Gaussian Filter یا Bilateral Filter استفاده کنیم. در این مرحله می توانیم پس از استفاده از یک فیلتر گاوسی، از adaptiveThreshold که تازه در درس آموختیم نیز استفاده کرده و سپس روی آن fastNlMeansDenoising هم بزنیم تا تصویر باینری داشته باشیم و لبه یابی دقیقتر انجام شود. با این روش، در انتهای این مرحله تصویر زیر را خواهیم داشت.



فیلتر Bilateral Filter نسخه پیشرفته فیلتر گاوسی است که برای صاف کردن تصاویر و کاهش نویز و در عین حال حفظ لبه ها استفاده می شود. در رویکرد دوم می توانیم از این فیلتر استفاده کرده و تنها blurring انجام دهیم و از thresholding صرف نظر کنیم. در این صورت نتیجه به صورت زیر است:



3- تشخیص لبه: از لبه یاب Canny استفاده می‌کنیم. با توجه به اینکه از چه رویکردی در مرحله قبل استفاده می‌کنیم باید threshold ها را کمی متفاوت تنظیم کنیم.

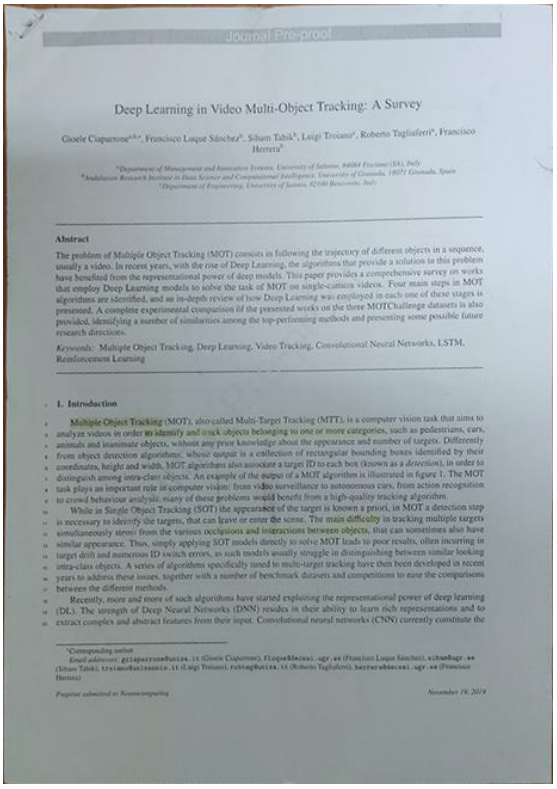
نتیجه با رویکرد اول:

نتیجه با رویکرد دوم:

4- تشخیص رئوس: از تابع findContours در OpenCV استفاده می‌کنیم که تمام نقاط پیوسته (در امتداد لبه ها) که رنگ یا شدت یکسانی دارند را باهم join می‌کند. چون می‌خواهیم در آخر تنها 4 نقطه گوشه برگه را ذخیره کنیم، Contour Approximation Method را CHAIN_APPROX_SIMPLE قرار می‌دهیم.

برای محاسبه بزرگترین contour در تصویر، روی contour های پیدا شده loop می‌زنیم و برای هر contour، از arclength و approxPolyDP استفاده می‌کنیم که بسته به دقتی که ما مشخص می کنیم، یک شکل کانتور را به شکل دیگری با تعداد رئوس کمتر تقریب می‌زند. اگر len(approx)، 4 باشد، isContourConvex برای آن True باشد و contourArea برای آن از مساحت تقریبی خود تصویر کوچکتر، و از بزرگترین erea ای که تا الان پیدا شده بزرگتر باشد، آنگاه contour را برابر این contour قرار می‌دهیم. در نهایت بزرگترین contour با همین روش پیدا می‌شود. نتیجه روی تصویر:

5- نگاشت دورنما و برش: ابتدا vertices را با استفاده از تابع reorder، به ترتیب بالا چپ، بالا راست، پایین راست و پایین چپ، مرتب می‌کنیم. width تصویر جدید را محاسبه می‌کنیم، که حداکثر فاصله بین مختصات x پایین راست و پایین چپ یا مختصات x بالا راست و بالا چپ است. همچنین ارتفاع تصویر جدید را محاسبه می‌کنیم، که حداکثر فاصله بین مختصات y بالا راست و پایین راست یا مختصات y بالا چپ و پایین چپ است. از تابع getPerspectiveTransform برای پیدا کردن تبدیلی استفاده می‌کنیم که منطقه‌ای را که شناسایی کرده‌ایم به یک مستطیل که به راحتی می‌توانیم آن را برش دهیم، map می‌کند. سپس باید از cv2.warpPerspective برای اعمال این تبدیل و برش منطقه هدف استفاده کنیم. نتیجه تصویر crop شده:



Journal Pre-proof

Deep Learning in Video Multi-Object Tracking: A Survey

Gioele Ciaparrone[a,b,*], Francisco Luque Sánchez[b], Siham Tabik[b], Luigi Troiano[c], Roberto Tagliaferri[a], Francisco Herrera[b]

[a]Department of Management and Innovation Systems, University of Salerno, 84084 Fisciano (SA), Italy
[b]Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, 18071 Granada, Spain
[c]Department of Engineering, University of Sannio, 82100 Benevento, Italy

Abstract

The problem of Multiple Object Tracking (MOT) consists in following the trajectory of different objects in a sequence, usually a video. In recent years, with the rise of Deep Learning, the algorithms that provide a solution to this problem have benefited from the representational power of deep models. This paper provides a comprehensive survey on works that employ Deep Learning models to solve the task of MOT on single-camera videos. Four main steps in MOT algorithms are identified, and an in-depth review of how Deep Learning was employed in each one of these stages is presented. A complete experimental comparison of the presented works on the three MOTChallenge datasets is also provided, identifying a number of similarities among the top-performing methods and presenting some possible future research directions.

Keywords: Multiple Object Tracking, Deep Learning, Video Tracking, Convolutional Neural Networks, LSTM, Reinforcement Learning

1. Introduction

Multiple Object Tracking (MOT), also called Multi-Target Tracking (MTT), is a computer vision task that aims to analyze videos in order to identify and track objects belonging to one or more categories, such as pedestrians, cars, animals and inanimate objects, without any prior knowledge about the appearance and number of targets. Differently from object detection algorithms, whose output is a collection of rectangular bounding boxes identified by their coordinates, height and width, MOT algorithms also associate a target ID to each box (known as a *detection*), in order to distinguish among intra-class objects. An example of the output of a MOT algorithm is illustrated in figure 1. The MOT task plays an important role in computer vision: from video surveillance to autonomous cars, from action recognition to crowd behaviour analysis, many of these problems would benefit from a high-quality tracking algorithm.

While in Single Object Tracking (SOT) the appearance of the target is known a priori, in MOT a detection step is necessary to identify the targets, that can leave or enter the scene. The main difficulty in tracking multiple targets simultaneously stems from the various occlusions and interactions between objects, that can sometimes also have similar appearance. Thus, simply applying SOT models directly to solve MOT leads to poor results, often incurring in target drift and numerous ID switch errors, as such models usually struggle in distinguishing between similar looking intra-class objects. A series of algorithms specifically tuned to multi-target tracking have then been developed in recent years to address these issues, together with a number of benchmark datasets and competitions to ease the comparisons between the different methods.

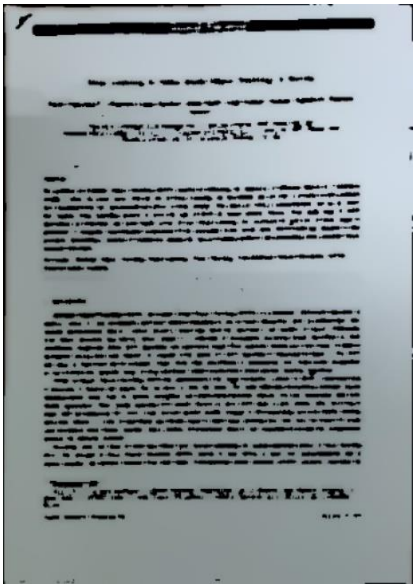
Recently, more and more of such algorithms have started exploiting the representational power of deep learning (DL). The strength of Deep Neural Networks (DNN) resides in their ability to learn rich representations and to extract complex and abstract features from their input. Convolutional neural networks (CNN) currently constitute the

*Corresponding author
Email addresses: gciaparrone@unisa.it (Gioele Ciaparrone), fluque@decsai.ugr.es (Francisco Luque Sánchez), stabik@ugr.es (Siham Tabik), ltroiano@unisannio.it (Luigi Troiano), robtag@unisa.it (Roberto Tagliaferri), herrera@decsai.ugr.es (Francisco Herrera)

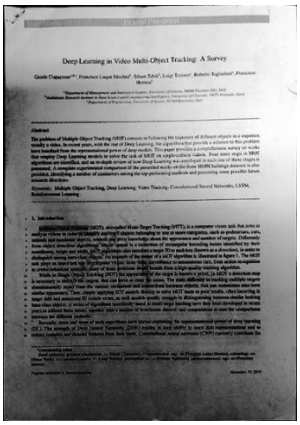
Preprint submitted to Neurocomputing *November 19, 2019*

6- مرحله آخر، بهبود تصویر: برای بهبود تصویر کراپ شده، روش های زیادی را جستجو و امتحان کردم، که کد آن ها در انتهای نوت بوک موجود است. برخی از آن ها نتیجه قابل قبولی نداشتند. برای مثال:

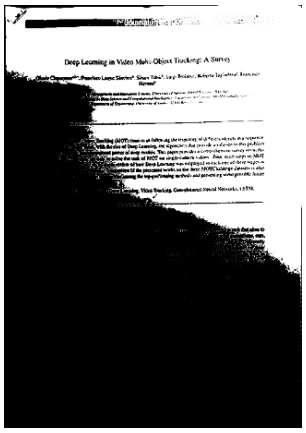
1. روش cartooning: این روش برای این تصویر نتیجه خوبی نداشت اما اگر عکس در شرایط بهتری گرفته شود می‌تواند نتیجه زیبایی داشته باشد.



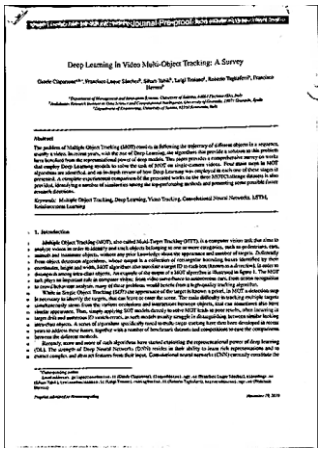
2. histogram equalization: نتیجه قابل قبول نبود.



3. thresholding: نتیجه برای این تصویر که نور آن بد بود اصلا قابل قبول نبود.

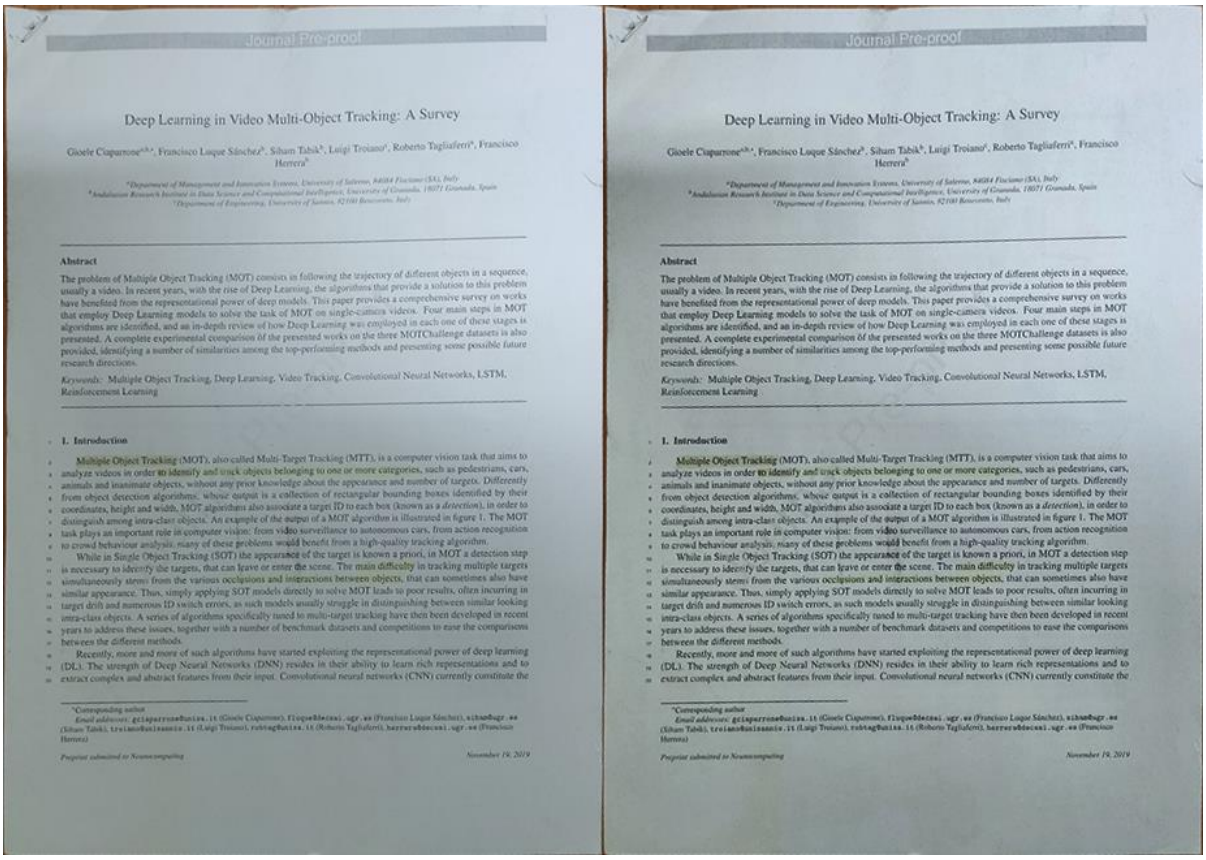


4. نتیجه adaptive thresholding: این روش از thresholding معمولی پیشرفته تر است و معمولا کار آمد است. اگر تصویر کمی در شرایط بهتری گرفته شود می‌تواند بسیار عالی عمل کند. در ادامه می‌بینیم برای تصویر دیگری که ورودی می‌دهم خیلی خوب عمل می‌کند.



روش های بهتر برای این تصویر:

5. تبدیل فضا به LAB و اعمال clahe (مقایسه با تصویر crop شده):



Journal Pre-proof

Deep Learning in Video Multi-Object Tracking: A Survey

Gioele Ciaparrone[a,b,*], Francisco Luque Sánchez[b], Siham Tabik[b], Luigi Troiano[c], Roberto Tagliaferri[a], Francisco Herrera[b]

[a]Department of Management and Innovation Systems, University of Salerno, 84084 Fisciano (SA), Italy
[b]Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, 18071 Granada, Spain
[c]Department of Engineering, University of Sannio, 82100 Benevento, Italy

Abstract

The problem of Multiple Object Tracking (MOT) consists in following the trajectory of different objects in a sequence, usually a video. In recent years, with the rise of Deep Learning, the algorithms that provide a solution to this problem have benefited from the representational power of deep models. This paper provides a comprehensive survey on works that employ Deep Learning models to solve the task of MOT on single-camera videos. Four main steps in MOT algorithms are identified, and an in-depth review of how Deep Learning was employed in each one of these stages is presented. A complete experimental comparison of the presented works on the three MOTChallenge datasets is also provided, identifying a number of similarities among the top-performing methods and presenting some possible future research directions.

Keywords: Multiple Object Tracking, Deep Learning, Video Tracking, Convolutional Neural Networks, LSTM, Reinforcement Learning

1. Introduction

Multiple Object Tracking (MOT), also called Multi-Target Tracking (MTT), is a computer vision task that aims to analyze videos in order to identify and track objects belonging to one or more categories, such as pedestrians, cars, animals and inanimate objects, without any prior knowledge about the appearance and number of targets. Differently from object detection algorithms, whose output is a collection of rectangular bounding boxes identified by their coordinates, height and width, MOT algorithms also associate a target ID to each box (known as a *detection*), in order to distinguish among intra-class objects. An example of the output of a MOT algorithm is illustrated in figure 1. The MOT task plays an important role in computer vision: from video surveillance to autonomous cars, from action recognition to crowd behaviour analysis, many of these problems would benefit from a high-quality tracking algorithm.

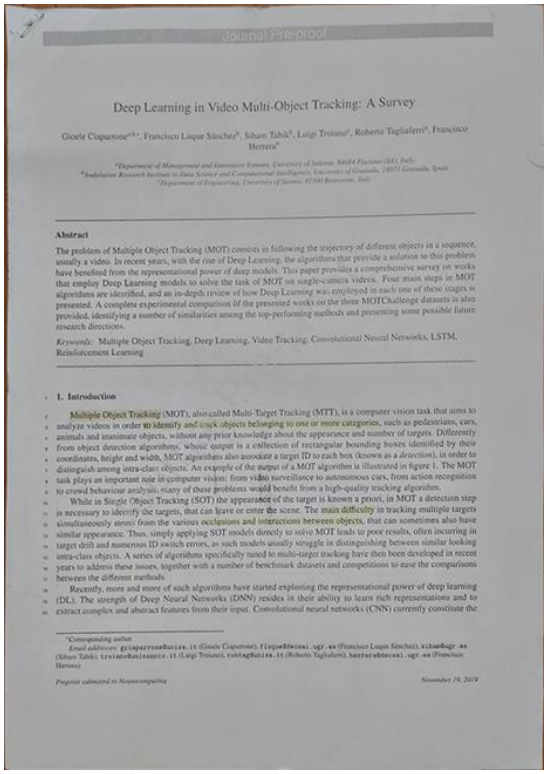
While in Single Object Tracking (SOT) the appearance of the target is known a priori, in MOT a detection step is necessary to identify the targets, that can leave or enter the scene. The main difficulty in tracking multiple targets simultaneously stems from the various occlusions and interactions between objects, that can sometimes also have similar appearance. Thus, simply applying SOT models directly to solve MOT leads to poor results, often incurring in target drift and numerous ID switch errors, as such models usually struggle in distinguishing between similar looking intra-class objects. A series of algorithms specifically tuned to multi-target tracking have then been developed in recent years to address these issues, together with a number of benchmark datasets and competitions to ease the comparisons between the different methods.

Recently, more and more of such algorithms have started exploiting the representational power of deep learning (DL). The strength of Deep Neural Networks (DNN) resides in their ability to learn rich representations and to extract complex and abstract features from their input. Convolutional neural networks (CNN) currently constitute the

*Corresponding author
Email addresses: gciaparrone@unisa.it (Gioele Ciaparrone), fluque@decsai.ugr.es (Francisco Luque Sánchez), siham@ugr.es (Siham Tabik), troiano@unisannio.it (Luigi Troiano), roberto@unisa.it (Roberto Tagliaferri), herrera@decsai.ugr.es (Francisco Herrera)

Preprint submitted to Neurocomputing November 19, 2019

6. روش تقسیم بر فیلتر گاوسی: ابتدا، با اعمال یک فیلتر محو بسیار قوی بر روی تصویر اصلی، یک تصویر "flat-field "fake تولید می کنیم. سپس تصویر crop شده را در میانگین F ضرب می کنیم و تصویر حاصل را بر F (پیکسل به پیکسل) تقسیم می‌کنیم تا تصویر اصلاح شده C بدست آید.
ضرب فقط برای حفظ روشنایی کلی است، اما بیشتر تغییرات در تقسیم اتفاق می افتد.
در تصویر اصلاح شده بسیاری از نور و رنگ های ناخواسته در مقیاس بزرگ حذف شده اند (مثل سایه ها).



Journal Pre-proof

Deep Learning in Video Multi-Object Tracking: A Survey

Gioele Ciaparrone[a,b,*], Francisco Luque Sánchez[b], Siham Tabik[b], Luigi Troiano[c], Roberto Tagliaferri[a], Francisco Herrera[b]

[a]Department of Management and Innovation Systems, University of Salerno, 84084 Fisciano (SA), Italy
[b]Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, 18071 Granada, Spain
[c]Department of Engineering, University of Sannio, 82100 Benevento, Italy

Abstract

The problem of Multiple Object Tracking (MOT) consists in following the trajectory of different objects in a sequence, usually a video. In recent years, with the rise of Deep Learning, the algorithms that provide a solution to this problem have benefited from the representational power of deep models. This paper provides a comprehensive survey on works that employ Deep Learning models to solve the task of MOT on single-camera videos. Four main steps in MOT algorithms are identified, and an in-depth review of how Deep Learning was employed in each one of these stages is presented. A complete experimental comparison of the presented works on the three MOTChallenge datasets is also provided, identifying a number of similarities among the top-performing methods and presenting some possible future research directions.

Keywords: Multiple Object Tracking, Deep Learning, Video Tracking, Convolutional Neural Networks, LSTM, Reinforcement Learning

1. Introduction

Multiple Object Tracking (MOT), also called Multi-Target Tracking (MTT), is a computer vision task that aims to analyze videos in order to identify and track objects belonging to one or more categories, such as pedestrians, cars, animals and inanimate objects, without any prior knowledge about the appearance and number of targets. Differently from object detection algorithms, whose output is a collection of rectangular bounding boxes identified by their coordinates, height and width, MOT algorithms also associate a target ID to each box (known as a *detection*), in order to distinguish among intra-class objects. An example of the output of a MOT algorithm is illustrated in figure 1. The MOT task plays an important role in computer vision: from video surveillance to autonomous cars, from action recognition to crowd behaviour analysis, many of these problems would benefit from a high-quality tracking algorithm.

While in Single Object Tracking (SOT) the appearance of the target is known a priori, in MOT a detection step is necessary to identify the targets, that can leave or enter the scene. The main difficulty in tracking multiple targets simultaneously stems from the various occlusions and interactions between objects, that can sometimes also have similar appearance. Thus, simply applying SOT models directly to solve MOT leads to poor results, often incurring in target drift and numerous ID switch errors, as such models usually struggle in distinguishing between similar looking intra-class objects. A series of algorithms specifically tuned to multi-target tracking have then been developed in recent years to address these issues, together with a number of benchmark datasets and competitions to ease the comparisons between the different methods.

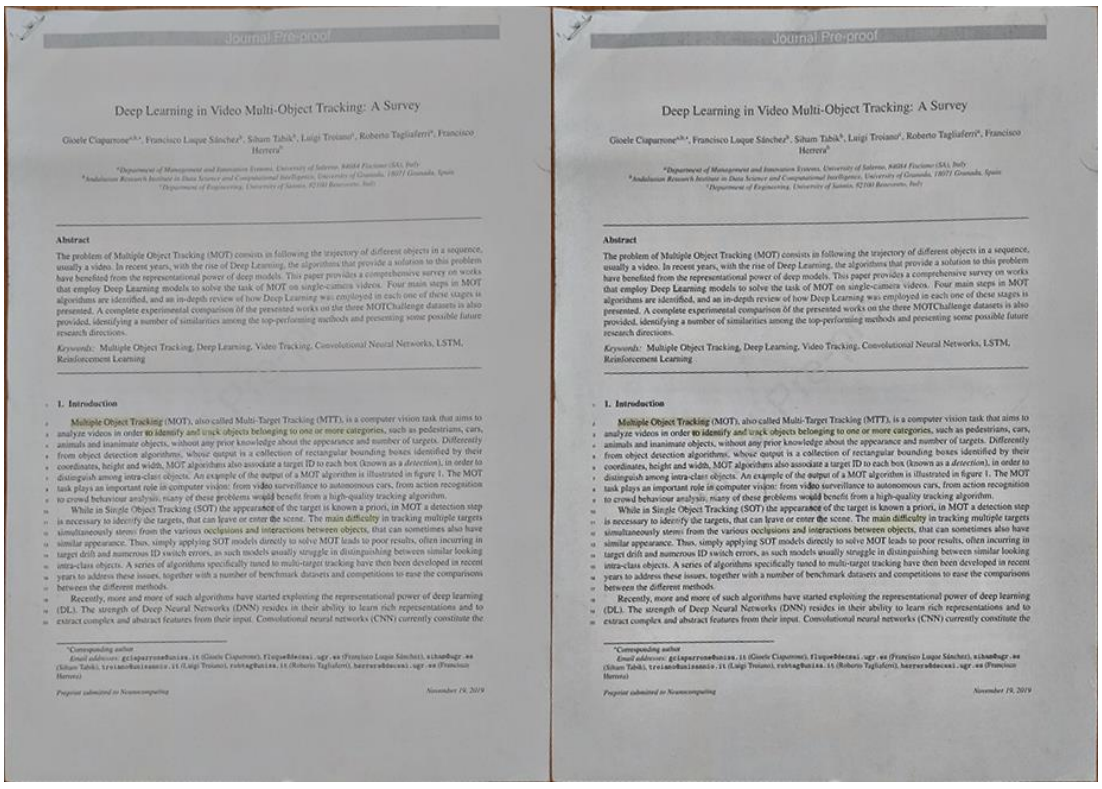
Recently, more and more of such algorithms have started exploiting the representational power of deep learning (DL). The strength of Deep Neural Networks (DNN) resides in their ability to learn rich representations and to extract complex and abstract features from their input. Convolutional neural networks (CNN) currently constitute the

*Corresponding author
Email addresses: gciaparrone@unisa.it (Gioele Ciaparrone), fluque@decsai.ugr.es (Francisco Luque Sánchez), siham@ugr.es (Siham Tabik), troiano@unisannio.it (Luigi Troiano), robtag@unisa.it (Roberto Tagliaferri), herrera@decsai.ugr.es (Francisco Herrera)

Preprint submitted to Neurocomputing *November 19, 2019*

می‌بینیم که سایه بزرگ پایین تصویر حذف شده است.

7. ترکیب روش بالا و CLAHE روی فضای رنگی LAB:



Journal Pre-proof

Deep Learning in Video Multi-Object Tracking: A Survey

Gioele Ciaparrone[a,b,*], Francisco Luque Sánchez[b], Siham Tabik[b], Luigi Troiano[c], Roberto Tagliaferri[a], Francisco Herrera[b]

[a]Department of Management and Innovation Systems, University of Salerno, 84084 Fisciano (SA), Italy
[b]Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, 18071 Granada, Spain
[c]Department of Engineering, University of Sannio, 82100 Benevento, Italy

Abstract

The problem of Multiple Object Tracking (MOT) consists in following the trajectory of different objects in a sequence, usually a video. In recent years, with the rise of Deep Learning, the algorithms that provide a solution to this problem have benefited from the representational power of deep models. This paper provides a comprehensive survey on works that employ Deep Learning models to solve the task of MOT on single-camera videos. Four main steps in MOT algorithms are identified, and an in-depth review of how Deep Learning was employed in each one of these stages is presented. A complete experimental comparison of the presented works on the three MOTChallenge datasets is also provided, identifying a number of similarities among the top-performing methods and presenting some possible future research directions.

Keywords: Multiple Object Tracking, Deep Learning, Video Tracking, Convolutional Neural Networks, LSTM, Reinforcement Learning

1. Introduction

Multiple Object Tracking (MOT), also called Multi-Target Tracking (MTT), is a computer vision task that aims to analyze videos in order to identify and track objects belonging to one or more categories, such as pedestrians, cars, animals and inanimate objects, without any prior knowledge about the appearance and number of targets. Differently from object detection algorithms, whose output is a collection of rectangular bounding boxes identified by their coordinates, height and width, MOT algorithms also associate a target ID to each box (known as a *detection*), in order to distinguish among intra-class objects. An example of the output of a MOT algorithm is illustrated in figure 1. The MOT task plays an important role in computer vision: from video surveillance to autonomous cars, from action recognition to crowd behaviour analysis, many of these problems would benefit from a high-quality tracking algorithm.

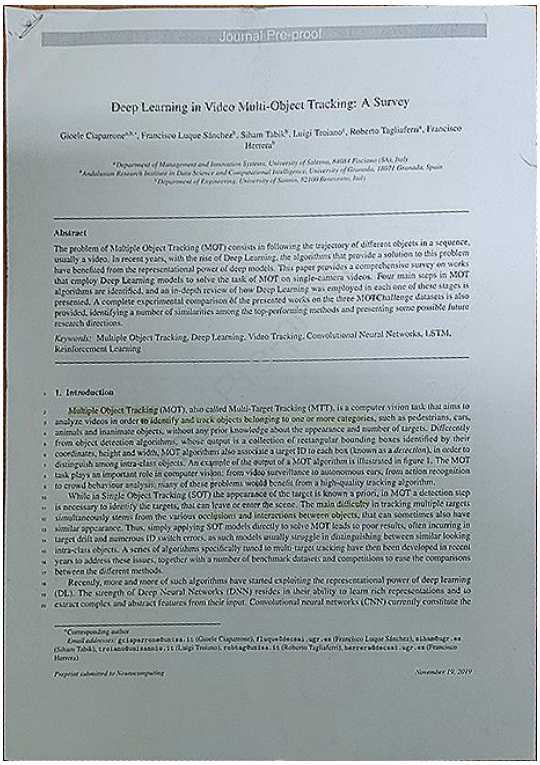
While in Single Object Tracking (SOT) the appearance of the target is known a priori, in MOT a detection step is necessary to identify the targets, that can leave or enter the scene. The main difficulty in tracking multiple targets simultaneously stems from the various occlusions and interactions between objects, that can sometimes also have similar appearance. Thus, simply applying SOT models directly to solve MOT leads to poor results, often incurring in target drift and numerous ID switch errors, as such models usually struggle in distinguishing between similar looking intra-class objects. A series of algorithms specifically tuned to multi-target tracking have then been developed in recent years to address these issues, together with a number of benchmark datasets and competitions to ease the comparisons between the different methods.

Recently, more and more of such algorithms have started exploiting the representational power of deep learning (DL). The strength of Deep Neural Networks (DNN) resides in their ability to learn rich representations and to extract complex and abstract features from their input. Convolutional neural networks (CNN) currently constitute the

*Corresponding author
Email addresses: gciaparrone@unisa.it (Gioele Ciaparrone), fluque@decsai.ugr.es (Francisco Luque Sánchez), siham@ugr.es (Siham Tabik), troiano@unisannio.it (Luigi Troiano), robtag@unisa.it (Roberto Tagliaferri), herrera@decsai.ugr.es (Francisco Herrera)

Preprint submitted to Neurocomputing *November 19, 2019*

8. Sharp کردن تصویر:



Journal Pre-proof

Deep Learning in Video Multi-Object Tracking: A Survey

Gioele Ciaparrone[a,b,*], Francisco Luque Sánchez[b], Siham Tabik[b], Luigi Troiano[c], Roberto Tagliaferri[a], Francisco Herrera[b]

[a]Department of Management and Innovation Systems, University of Salerno, 84084 Fisciano (SA), Italy
[b]Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, 18071 Granada, Spain
[c]Department of Engineering, University of Sannio, 82100 Benevento, Italy

Abstract

The problem of Multiple Object Tracking (MOT) consists in following the trajectory of different objects in a sequence, usually a video. In recent years, with the rise of Deep Learning, the algorithms that provide a solution to this problem have benefited from the representational power of deep models. This paper provides a comprehensive survey on works that employ Deep Learning models to solve the task of MOT on single-camera videos. Four main steps in MOT algorithms are identified, and an in-depth review of how Deep Learning was employed in each one of these stages is presented. A complete experimental comparison of the presented works on the three MOTChallenge datasets is also provided, identifying a number of similarities among the top-performing methods and presenting some possible future research directions.

Keywords: Multiple Object Tracking, Deep Learning, Video Tracking, Convolutional Neural Networks, LSTM, Reinforcement Learning

1. Introduction

Multiple Object Tracking (MOT), also called Multi-Target Tracking (MTT), is a computer vision task that aims to analyze videos in order to identify and track objects belonging to one or more categories, such as pedestrians, cars, animals and inanimate objects, without any prior knowledge about the appearance and number of targets. Differently from object detection algorithms, whose output is a collection of rectangular bounding boxes identified by their coordinates, height and width, MOT algorithms also associate a target ID to each box (known as a *detection*), in order to distinguish among intra-class objects. An example of the output of a MOT algorithm is illustrated in figure 1. The MOT task plays an important role in computer vision: from video surveillance to autonomous cars, from action recognition to crowd behaviour analysis, many of these problems would benefit from a high-quality tracking algorithm.

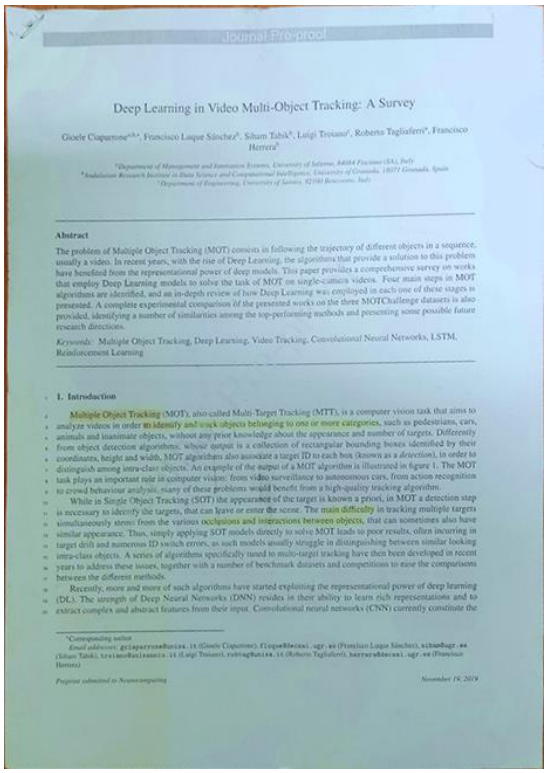
While in Single Object Tracking (SOT) the appearance of the target is known a priori, in MOT a detection step is necessary to identify the targets, that can leave or enter the scene. The main difficulty in tracking multiple targets simultaneously stems from the various occlusions and interactions between objects, that can sometimes also have similar appearance. Thus, simply applying SOT models directly to solve MOT leads to poor results, often incurring in target drift and numerous ID switch errors, as such models usually struggle in distinguishing between similar looking intra-class objects. A series of algorithms specifically tuned to multi-target tracking have then been developed in recent years to address these issues, together with a number of benchmark datasets and competitions to ease the comparisons between the different methods.

Recently, more and more of such algorithms have started exploiting the representational power of deep learning (DL). The strength of Deep Neural Networks (DNN) resides in their ability to learn rich representations and to extract complex and abstract features from their input. Convolutional neural networks (CNN) currently constitute the

*Corresponding author
Email addresses: gciaparrone@unisa.it (Gioele Ciaparrone), fluque@decsai.ugr.es (Francisco Luque Sánchez), siham@ugr.es (Siham Tabik), troiano@unisannio.it (Luigi Troiano), robtag@unisa.it (Roberto Tagliaferri), herrera@decsai.ugr.es (Francisco Herrera)

Preprint submitted to Neurocomputing *November 19, 2019*

9. بالا بردن saturation و value تصویر:



Journal Pre-proof

Deep Learning in Video Multi-Object Tracking: A Survey

Gioele Ciaparrone[a,b,*], Francisco Luque Sánchez[b], Siham Tabik[b], Luigi Troiano[c], Roberto Tagliaferri[a], Francisco Herrera[b]

[a]Department of Management and Innovation Systems, University of Salerno, 84084 Fisciano (SA), Italy
[b]Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, 18071 Granada, Spain
[c]Department of Engineering, University of Sannio, 82100 Benevento, Italy

Abstract

The problem of Multiple Object Tracking (MOT) consists in following the trajectory of different objects in a sequence, usually a video. In recent years, with the rise of Deep Learning, the algorithms that provide a solution to this problem have benefited from the representational power of deep models. This paper provides a comprehensive survey on works that employ Deep Learning models to solve the task of MOT on single-camera videos. Four main steps in MOT algorithms are identified, and an in-depth review of how Deep Learning was employed in each one of these stages is presented. A complete experimental comparison of the presented works on the three MOTChallenge datasets is also provided, identifying a number of similarities among the top-performing methods and presenting some possible future research directions.

Keywords: Multiple Object Tracking, Deep Learning, Video Tracking, Convolutional Neural Networks, LSTM, Reinforcement Learning

1. Introduction

Multiple Object Tracking (MOT), also called Multi-Target Tracking (MTT), is a computer vision task that aims to analyze videos in order to identify and track objects belonging to one or more categories, such as pedestrians, cars, animals and inanimate objects, without any prior knowledge about the appearance and number of targets. Differently from object detection algorithms, whose output is a collection of rectangular bounding boxes identified by their coordinates, height and width, MOT algorithms also associate a target ID to each box (known as a *detection*), in order to distinguish among intra-class objects. An example of the output of a MOT algorithm is illustrated in figure 1. The MOT task plays an important role in computer vision: from video surveillance to autonomous cars, from action recognition to crowd behaviour analysis, many of these problems would benefit from a high-quality tracking algorithm.

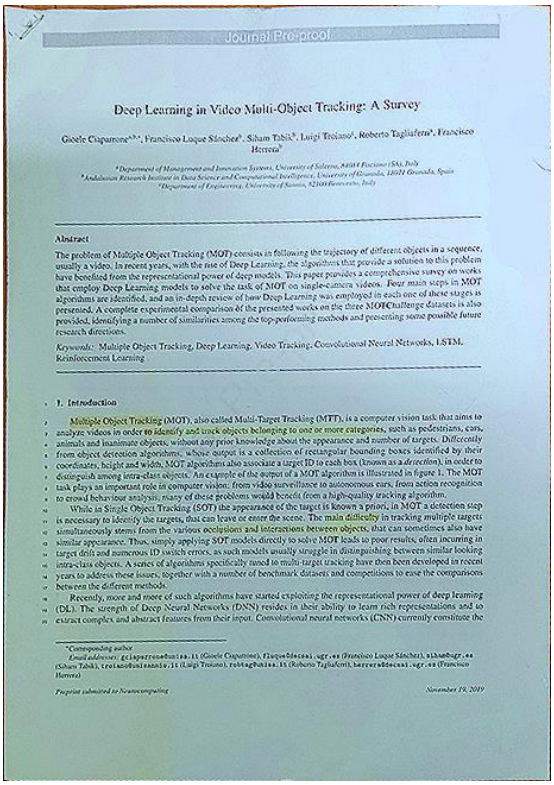
While in Single Object Tracking (SOT) the appearance of the target is known a priori, in MOT a detection step is necessary to identify the targets, that can leave or enter the scene. The main difficulty in tracking multiple targets simultaneously stems from the various occlusions and interactions between objects, that can sometimes also have similar appearance. Thus, simply applying SOT models directly to solve MOT leads to poor results, often incurring in target drift and numerous ID switch errors, as such models usually struggle in distinguishing between similar looking intra-class objects. A series of algorithms specifically tuned to multi-target tracking have then been developed in recent years to address these issues, together with a number of benchmark datasets and competitions to ease the comparisons between the different methods.

Recently, more and more of such algorithms have started exploiting the representational power of deep learning (DL). The strength of Deep Neural Networks (DNN) resides in their ability to learn rich representations and to extract complex and abstract features from their input. Convolutional neural networks (CNN) currently constitute the

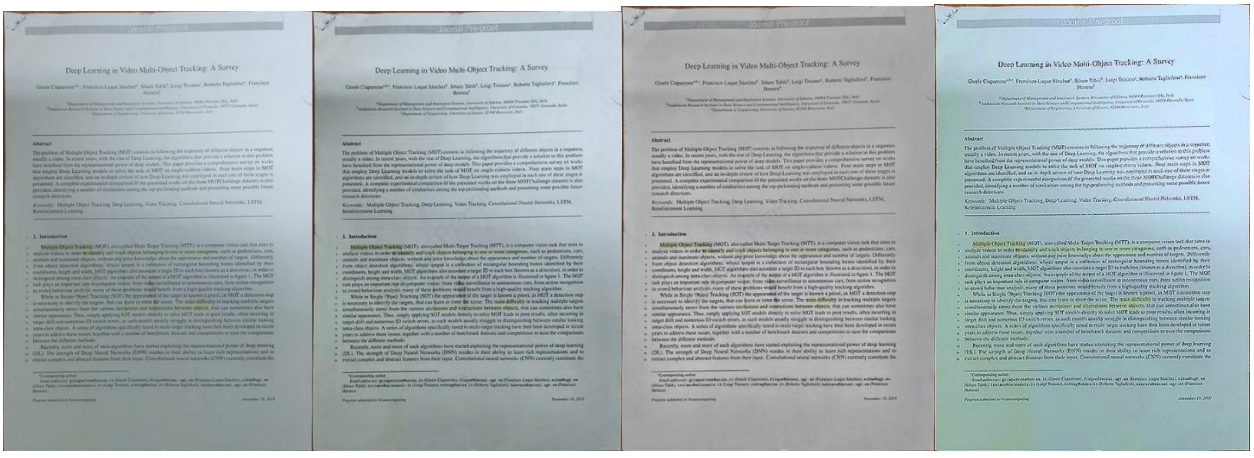
*Corresponding author
Email addresses: gciaparrone@unisa.it (Gioele Ciaparrone), fluque@decsai.ugr.es (Francisco Luque Sánchez), siham@ugr.es (Siham Tabik), troiano@unisannio.it (Luigi Troiano), robtag@unisa.it (Roberto Tagliaferri), herrera@decsai.ugr.es (Francisco Herrera)

Preprint submitted to Neurocomputing *November 19, 2019*

10. ترکیب روش Sharp کردن تصویر و بالا بردن saturation و value:

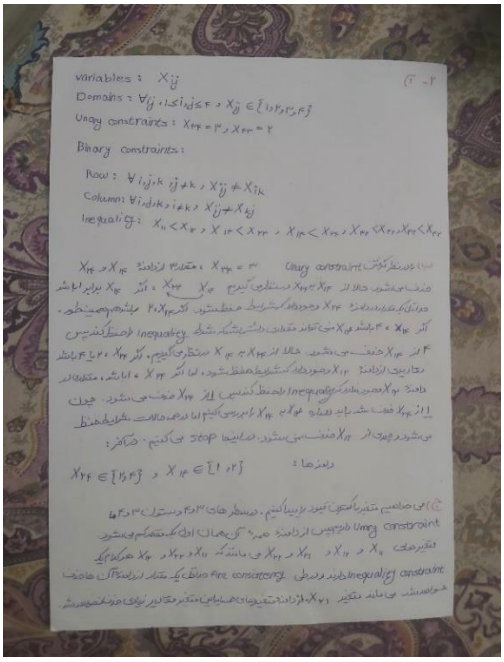


در کل، این روش ها را می‌توان به عنوان فیلتر های مختلف camscanner خود در نظر بگیریم که ممکن است هرکدام برای تصویر های مختلف خروجی های بهتر یا بدتری داشته باشند. بهترین خروجی ها برای این تصویر مربوط به روش های 4 و 6 و 8 است. مقایسه نتیجه این روش ها را در کنار تصویر crop شده اصلی می‌بینیم:

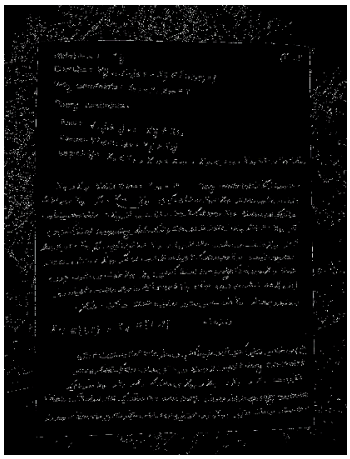


کد همه روش ها در نوت بوک موجود هست. اما در تابع enhance کد روش آخر را قرار دادم که نتیجه بهتری برای این تصویر داشت.

7- پارت امتیازی: برای امتحان کردن نحوه کارکرد برنامه تصویری که خودم گرفتم را به برنامه می‌دهم.
این تصویر در پوشه images با نام example2.jpg ذخیره شده است.



نتیجه مراحل تشخیص لبه و تشخیص رئوس برای این تصویر:

می‌بینیم که نقاط گوشه برگه به دلیل رنگ و نویز زیاد بک گراند پیدا نشدند. و در واقع همان نقاط اولیه (نقاط گوشه خود تصویر) که در ابتدا به آن مقداردهی کرده بودم را دارد.
ابتدا سعی کردم با قویتر و ضعیفتر کردن فیلتر گاوسی یا bilateral و عوض کردن threshold های canny خروجی را بهبود دهم اما باز هم چارچوب صفحه درست لبه یابی نشد و نقاط گوشه برگه را تشخیص نداد.

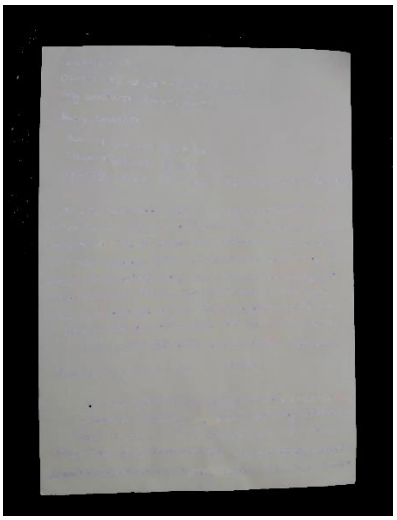
تغییر پیشنهادی در الگوریتم:

می‌توانیم از پردازش های مورفولوژی که به تازگی در درس یاد گرفتیم استفاده کنیم. (مربوط به بحث image segmentation). با استفاده از تابع cv2.morphologyEx، با انجام عملیاتی مانند erosion و dilation، می توانیم تبدیل های مورفولوژیکی پیشرفته ای را انجام دهیم. در اینجا، ما عملیات closing را انجام می‌دهیم.

عملگر بسته (closing) برای حذف حفره های کوچک و هموار کردن محیط نواحی تعریف شده است. این عملگر ناحیه های سیاه که در احاطه پیکسل های سفید هستند را حذف میکند. با این کار محتویات برگه را حذف کرده و در تصویر به یک صفحه برگه سفید خالی می‌رسیم.

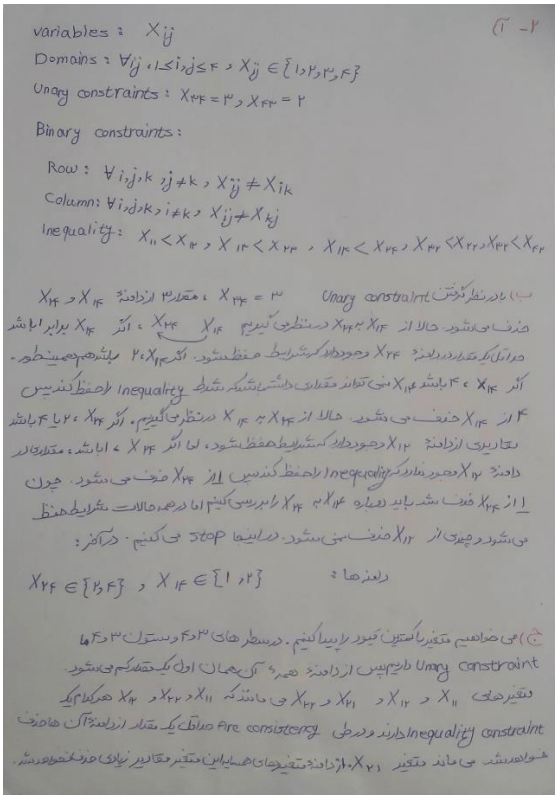
گام بعدی حذف پس زمینه تصویر است. از GrabCut در OpenCV استفاده می کنیم. برای این کار نیاز به یک کادر محدود در اطراف جسم داریم، سپس هر چیزی که خارج از این box است، به عنوان پس زمینه در نظر گرفته می شود.

GrabCut به طور خودکار ما را از شر تمام پس‌زمینه‌ها خلاص می‌کند، و تنها برگه ما باقی می‌ماند. پس 20 پیکسل گوشه را به عنوان پس‌زمینه می‌گیریم و GrabCut به‌طور خودکار پیش‌زمینه و پس‌زمینه را تعیین می‌کند و تنها برگه را برای ما باقی می‌گذارد. نتیجه:



پس از این مرحله همه مراحل قبلی را به ترتیب مانند قبل انجام می‌دهیم. در واقع این مرحله به عنوان مرحله اول اضافه شده است. حالا کار لبه یاب بسیار راحت تر است. نتیجه نقطه یابی:

تصویر crop شده:



بهترین نتیجه های بهبود تصویر که پیاده سازی کردم، برای این تصویر:

:Adaptive Thresholding

۲- آ)

variables : X_{ij}

Domains : $\forall ij \; , 1 \le i,j \le 4 \; , X_{ij} \in \{1,2,3,4\}$

Unary constraints : $X_{34} = 3 \; , X_{43} = 2$

Binary constraints :

Row : $\forall i,j,k \; , j \neq k \; , X_{ij} \neq X_{ik}$

Column : $\forall i,j,k \; , i \neq k \; , X_{ij} \neq X_{kj}$

Inequality : $X_{11} < X_{12} \; , X_{13} < X_{23} \; , X_{14} < X_{24} \; , X_{32} < X_{22} , X_{32} < X_{42}$

ب) با در نظر گرفتن Unary constraint $X_{34} = 3$ ، مقدار ۳ از دامنهٔ X_{14} و X_{24} حذف می‌شود. حالا از X_{14} به X_{24} در نظر می‌گیریم ($X_{14} \rightarrow X_{24}$) ؛ اگر X_{14} برابر ۱ باشد حداقل یک مقدار در دامنهٔ X_{24} وجود دارد که شرایط حفظ شود. اگر X_{14}، ۲ باشد هم همینطور. اگر X_{14} ، ۴ باشد X_{24} نمی‌تواند مقداری داشته باشد که شرط Inequality را حفظ کند پس ۴ از X_{14} حذف می‌شود. حالا از X_{24} به X_{14} در نظر می‌گیریم، اگر X_{24} ، ۲ یا ۴ باشد مقادیری از دامنهٔ X_{14} وجود دارد که شرایط حفظ شود، اما اگر X_{24} ، ۱ باشد، مقداری در دامنهٔ X_{14} وجود ندارد که Inequality را حفظ کند پس ۱ از X_{24} حذف می‌شود. چون ۱ از X_{24} حذف شد باید دوباره X_{14} به X_{24} را بررسی کنیم اما در همه حالات شرایط حفظ می‌شود و چیزی از X_{14} حذف نمی‌شود. در اینجا stop می‌کنیم. در آخر:

دامنه‌ها : $X_{14} \in \{1,2\} \; , X_{24} \in \{2,4\}$

ج) می‌خواهیم متغیر با کمترین قیود را پیدا کنیم. در سطرهای ۳ و ۴ و ستون ۳ و ۴ ها Unary constraint داریم پس از دامنهٔ همهٔ آن‌ها ۲ مقدار کم می‌شود. متغیرهای X_{11} و X_{12} و X_{21} و X_{22} می‌مانند که X_{11} و X_{22} و X_{12} هرکدام یک Inequality constraint دارند و در طی Are consistency حداقل یک مقدار از دامنهٔ آن‌ها حذف خواهد شد. می‌ماند متغیر X_{21} که از دامنهٔ متغیرهای همسایه‌اش متغیر مقادیر زیادی حذف خواهد شد.

:Cartooning

۲- آ)

variables : X_{ij}

Domains : $\forall ij \; , 1 \le i,j \le 4 \; , \; X_{ij} \in \{1,2,3,4\}$

Unary constraints : $X_{34} = 3 \; , \; X_{43} = 2$

Binary constraints :

Row : $\forall i,j,k \; , j \neq k \; , \; X_{ij} \neq X_{ik}$

Column : $\forall i,j,k \; , i \neq k \; , \; X_{ij} \neq X_{kj}$

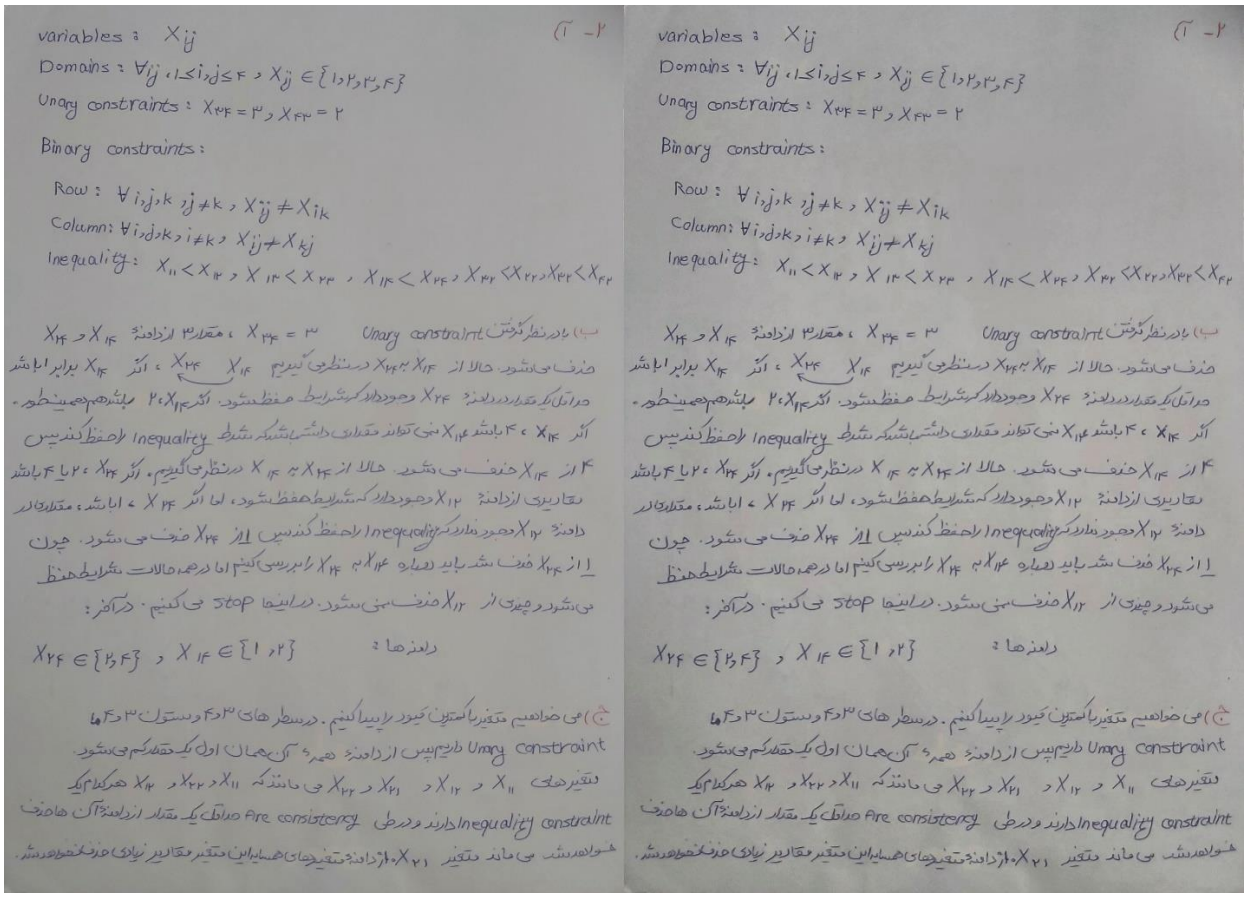
Inequality : $X_{11} < X_{12} \; , \; X_{13} < X_{23} \; , \; X_{14} < X_{24} \; , \; X_{32} < X_{22} \; , \; X_{32} < X_{42}$

ب) با در نظر گرفتن Unary constraint $X_{34} = 3$ ، مقدار ۳ از دامنهٔ X_{14} و X_{24} حذف می‌شود. حالا از X_{14} به X_{24} در نظر می‌گیریم ($X_{14} \rightarrow X_{24}$)، اگر X_{14} برابر ۱ باشد حداقل یک مقدار در دامنهٔ X_{24} وجود دارد که شرایط حفظ شود. اگر X_{14}، ۲ باشد هم همینطور. اگر X_{14} ، ۴ باشد X_{24} نمی‌تواند مقداری داشته باشد که شرط Inequality را حفظ کند پس ۴ از X_{14} حذف می‌شود. حالا از X_{24} به X_{14} در نظر می‌گیریم. اگر X_{24} ، ۲ یا ۴ باشد مقادیری از دامنهٔ X_{12} وجود دارد که شرایط حفظ شود، اما اگر X_{24} ، ۱ باشد، مقداری در دامنهٔ X_{12} وجود ندارد که Inequality را حفظ کند پس ۱ از X_{24} حذف می‌شود. چون ۱ از X_{24} حذف شد باید دوباره X_{14} به X_{24} را بررسی کنیم اما در همه حالات شرایط حفظ می‌شود و چیزی از X_{12} حذف نمی‌شود. در اینجا stop می‌کنیم. در آخر:

دامنه‌ها : $X_{14} \in \{1,2\} \; , \; X_{24} \in \{2,4\}$

ج) می‌خواهیم متغیر با کمترین قیود را پیدا کنیم. در سطرهای ۳ و ۴ و ستون ۳ و ۴ ما Unary constraint داریم پس از دامنهٔ همهٔ آن‌ها مقدار کم می‌شود. متغیرهای X_{11} و X_{12} و X_{21} و X_{22} می‌مانند که X_{11} و X_{22} و X_{12} هرکدام یک Inequality constraint دارند و در طی Are consistency حداقل یک مقدار از دامنهٔ آن‌ها حذف خواهد شد. می‌ماند متغیر X_{21} و از دامنهٔ متغیرهای همسایه این متغیر مقادیر زیادی حذف خواهد شد.

:CLAHE



منابع:

لینک

لینک

لینک

لینک

لینک