Automatic Text Identification, Recognition and Physical Address Generation from Unstructured Video

P Thaiseer (SC13B175)

Under the guidance of Dr. Gorthi R. K. S. S. Manyam and Dr. V. K. Dadhwal

Indian Institute of Space Science and Technology
Thiruvananthapuram - 695547

May 8, 2017

Outline

Importance of Topic

- Natural scene images / videos contains a lot of texts which conveys valuable information about that scene.

Importance of Topic

- Natural scene images / videos contains a lot of texts which conveys valuable information about that scene.
- If we can connect these information with the location data (latitude and longitude) of image / video, these can be very useful in many applications:

Importance of Topic

- Natural scene images / videos contains a lot of texts which conveys valuable information about that scene.
- If we can connect these information with the location data (latitude and longitude) of image / video, these can be very useful in many applications:
  - Track a video on a map and add informations on to the map from extracted information.

Dataset

- A 38 second video (16fps).

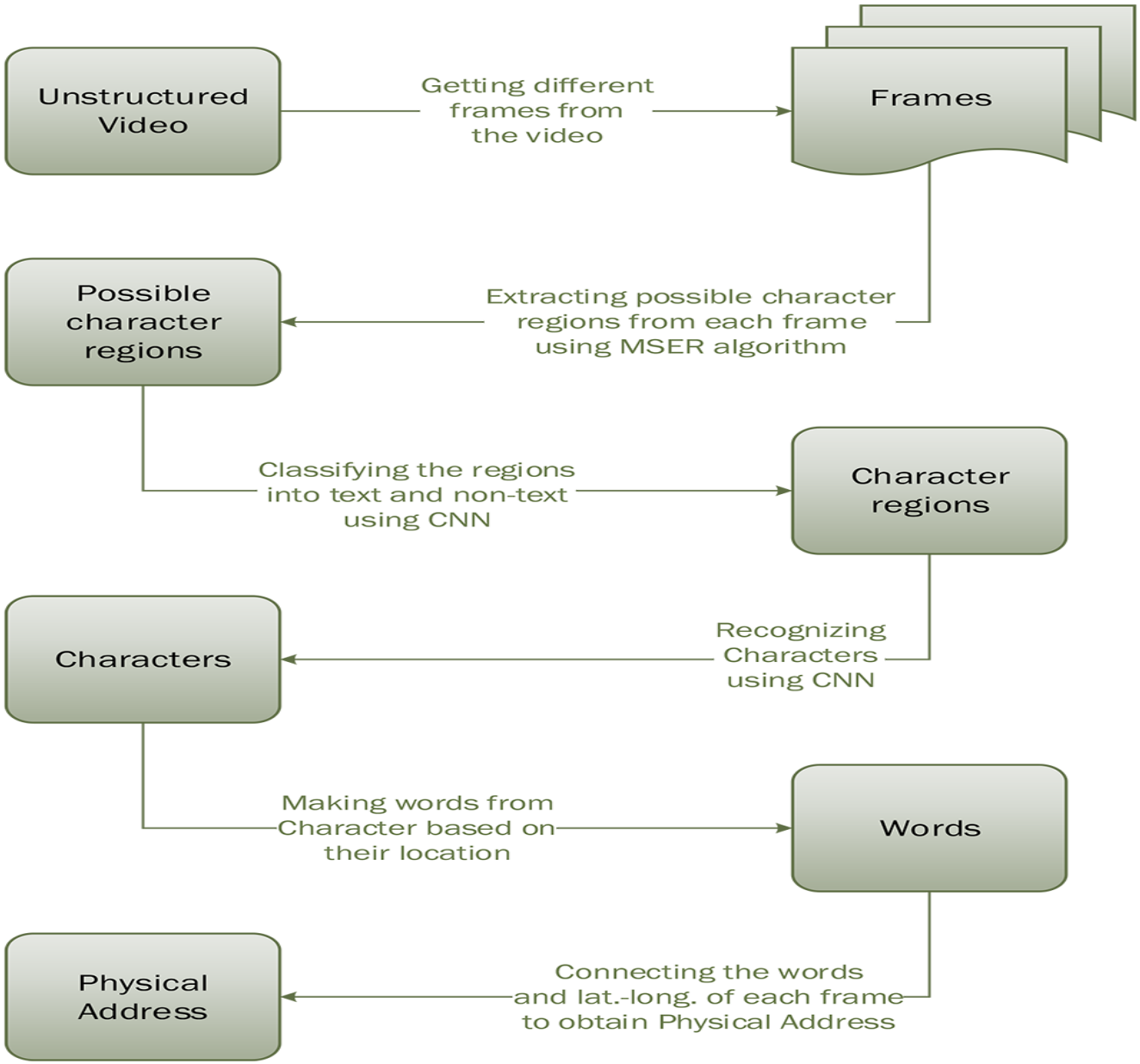
Dataset

- A 38 second video (16fps).

Dataset

- A 38 second video (16fps).
- Latitude and longitude of camera for each second.

Work-flow


Unstructured Video
Getting different frames from the video
Frames
Possible character regions
Extracting possible character regions from each frame using MSER algorithm
Classifying the regions into text and non-text using CNN
Character regions
Characters
Recognizing Characters using CNN
Making words from Character based on their location
Words
Physical Address
Connecting the words and lat.-long. of each frame to obtain Physical Address

Edge based Text Extraction

- Initial Frame

Edge based Text Extraction

- Text regions identified

Edge based Text Extraction

- Same method failed for few other frames:

Figure 1: Initial Image

Edge based Text Extraction

- Same method failed for few other frames:

Figure 1: Initial Image

Figure 2: Final Image

Edge based Text Extraction

- Same method failed for few other frames:

Figure 3: Initial Image

Edge based Text Extraction

- Same method failed for few other frames:

Figure 3: Initial Image

Figure 4: Final Image

New Approach

- Super-pixel segmentation.

New Approach

- Super-pixel segmentation.
- Quick shift image segmentation.

New Approach

- Super-pixel segmentation.
- Quick shift image segmentation.
- Finally:
  - Maximally stable extremal regions - MSERs [4]

Maximally stable extremal regions (MSERs)

- MSER algorithm extracts stable regions from an image by considering the change in area with respect to the change in intensity of a connected component defined by thresholding the image at different gray levels.

Maximally stable extremal regions (MSERs)

- MSER algorithm extracts stable regions from an image by considering the change in area with respect to the change in intensity of a connected component defined by thresholding the image at different gray levels.
- Why MSER?

Maximally stable extremal regions (MSERs)

- MSER algorithm extracts stable regions from an image by considering the change in area with respect to the change in intensity of a connected component defined by thresholding the image at different gray levels.
- Why MSER?
  - *Invariance* to affine transformation of image intensities.

Maximally stable extremal regions (MSERs)

- MSER algorithm extracts stable regions from an image by considering the change in area with respect to the change in intensity of a connected component defined by thresholding the image at different gray levels.

- Why MSER?
  - *Invariance* to affine transformation of image intensities.
  - *Stability*: Only regions whose support is nearly the same over a range of thresholds is selected.

Maximally stable extremal regions (MSERs)

- MSER algorithm extracts stable regions from an image by considering the change in area with respect to the change in intensity of a connected component defined by thresholding the image at different gray levels.

- Why MSER?
  - *Invariance* to affine transformation of image intensities.
  - *Stability*: Only regions whose support is nearly the same over a range of thresholds is selected.
  - *Multi-scale* detection, both fine and large structure is detected.

Maximally stable extremal regions (MSERs)

- MSER algorithm extracts stable regions from an image by considering the change in area with respect to the change in intensity of a connected component defined by thresholding the image at different gray levels.
- MSER is based on the idea of taking regions which stay nearly the same through a wide range of thresholds.

Maximally stable extremal regions (MSERs)

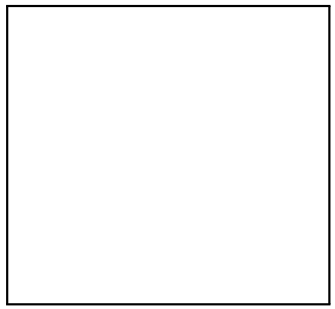
- MSER algorithm extracts stable regions from an image by considering the change in area with respect to the change in intensity of a connected component defined by thresholding the image at different gray levels.
- MSER is based on the idea of taking regions which stay nearly the same through a wide range of thresholds.
  - All the pixels above a given threshold are white and all those below or equal are black.

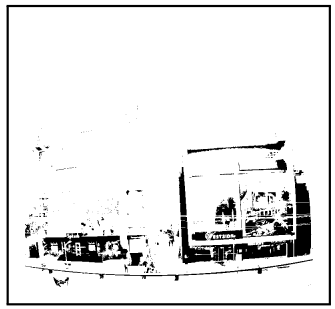
Maximally stable extremal regions (MSERs)

- MSER algorithm extracts stable regions from an image by considering the change in area with respect to the change in intensity of a connected component defined by thresholding the image at different gray levels.
- MSER is based on the idea of taking regions which stay nearly the same through a wide range of thresholds.
  - All the pixels above a given threshold are white and all those below or equal are black.
  - If we are shown a sequence of thresholded images I_t with t corresponding to threshold t, we would see first a white image, then black spots starts appearing then grow larger. These will eventually merge, until the whole image is black.

Maximally stable extremal regions (MSERs)


(a) Initial


(b) t = 10


(c) t = 50


(d) t = 60


(e) t = 80


(f) t = 70


(g) t = 90


(h) t = 100

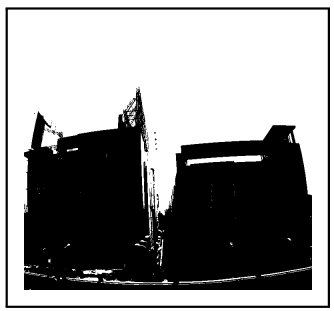

(i) t = 110


(j) t = 120

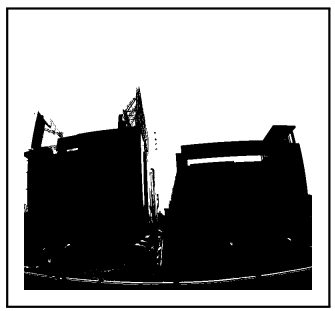

(k) t = 130


(l) t = 140

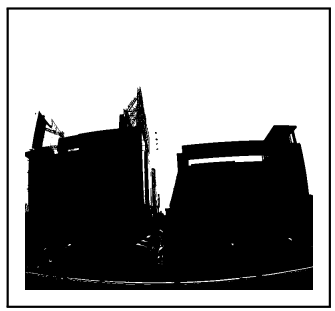
Maximally stable extremal regions (MSERs)



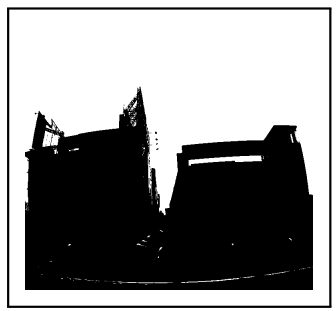
(m) t = 150



(n) t = 160

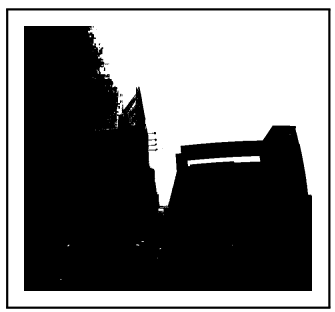


(o) t = 170



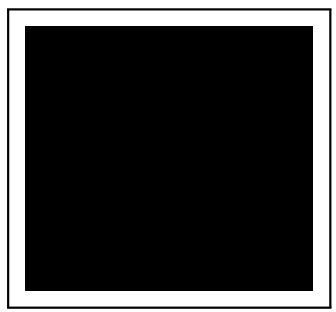
(p) t = 180

(q) t = 190



(r) t = 200

(s) t = 220



(t) t = 255

Maximally stable extremal regions (MSERs)

- The set of all connected components in these sequence of images is the set of all extremal regions.

Maximally stable extremal regions (MSERs)

- The set of all connected components in these sequence of images is the set of all extremal regions.
- In extremal regions, pixel values are either strictly darker or strictly brighter than those on the boundary.

Maximally stable extremal regions (MSERs)

- The set of all connected components in these sequence of images is the set of all extremal regions.
- In extremal regions, pixel values are either strictly darker or strictly brighter than those on the boundary.
- A region is considered stable if its area changes only slightly with the change of threshold t.

MSERs - Algorithm [4]

- Pixels are sorted by intensity. (Order: n)

MSERs - Algorithm [4]

- Pixels are sorted by intensity. (Order: n)
- After sorting, pixels are placed in the image (either in decreasing or increasing order).

MSERs - Algorithm [4]

- Pixels are sorted by intensity. (Order: n)
- After sorting, pixels are placed in the image (either in decreasing or increasing order).
- The list of connected components and their areas is tracked using the efficient union-find algorithm. (Order: nlog(log(n)))

MSERs - Algorithm [4]

- Pixels are sorted by intensity. (Order: n)
- After sorting, pixels are placed in the image (either in decreasing or increasing order).
- The list of connected components and their areas is tracked using the efficient union-find algorithm. (Order: nlog(log(n)))
- The process produces a data structure storing the area of each connected component as a function of intensity.

MSERs - Algorithm [4]

- Pixels are sorted by intensity. (Order: n)
- After sorting, pixels are placed in the image (either in decreasing or increasing order).
- The list of connected components and their areas is tracked using the efficient union-find algorithm. (Order: nlog(log(n)))
- The process produces a data structure storing the area of each connected component as a function of intensity.
- Finally, intensity levels that are local minima of the rate of change of the area function are selected as thresholds producing maximally stable extremal regions.

MSERs - Observations

- MSER applied on simple images (ICDAR[1] 2015 dataset)

[1] ICDAR 2015. "Incidental Scene Text Dataset". In: *International Conference on Document Analysis and Recognition* (2015).

MSERs - Observations

- MSER applied on simple images (ICDAR[1] 2015 dataset)

Figure 5: MSERs initially identified on ICDAR[1] 2015 image

[1] ICDAR 2015. "Incidental Scene Text Dataset". In: *International Conference on Document Analysis and Recognition* (2015).

MSERs - Observations

- MSER applied on video frames

MSERs - Observations

- MSER applied on video frames

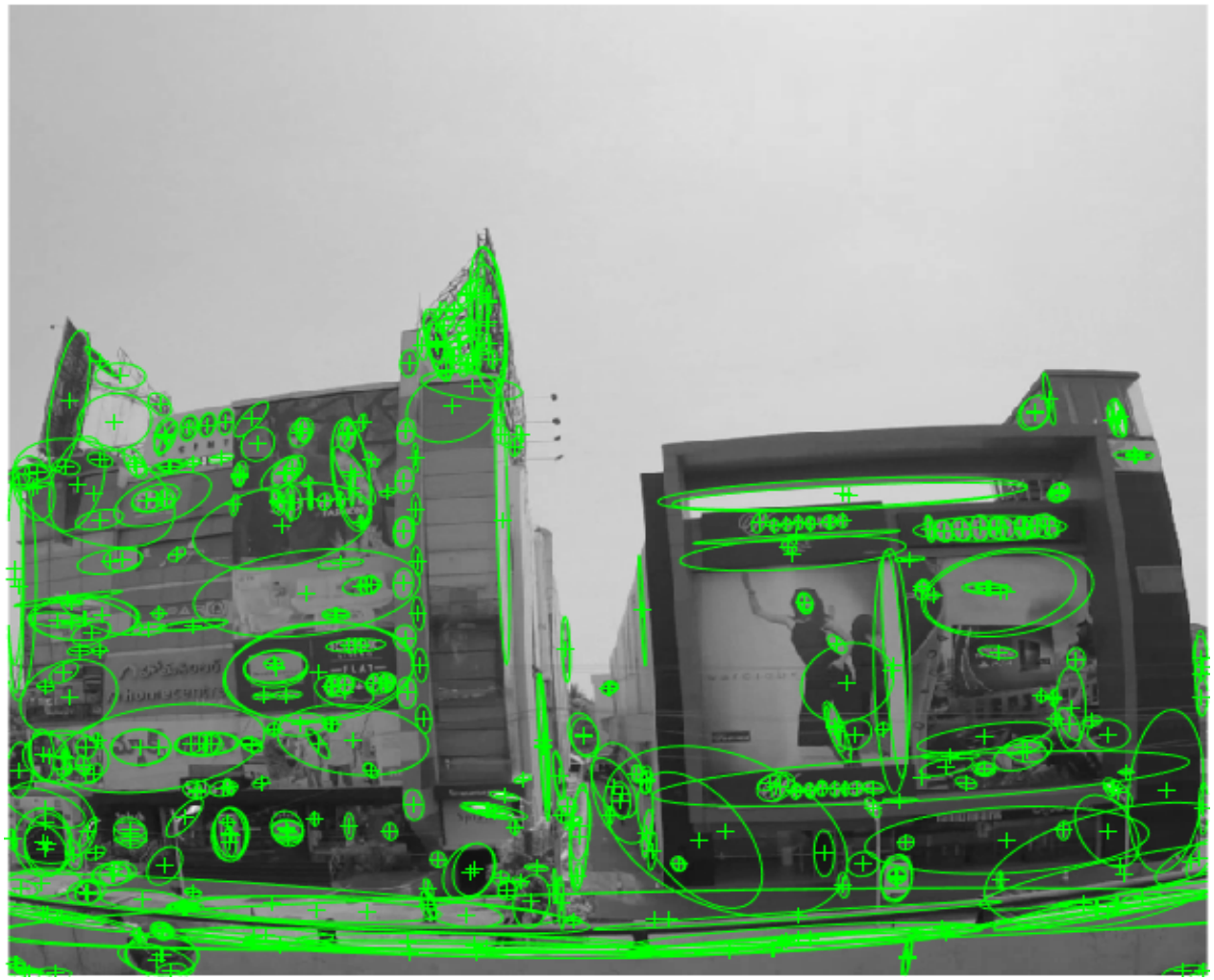


Figure 6: MSERs initially identified on video frame

MSERs - Observations

- MSER applied on video frames

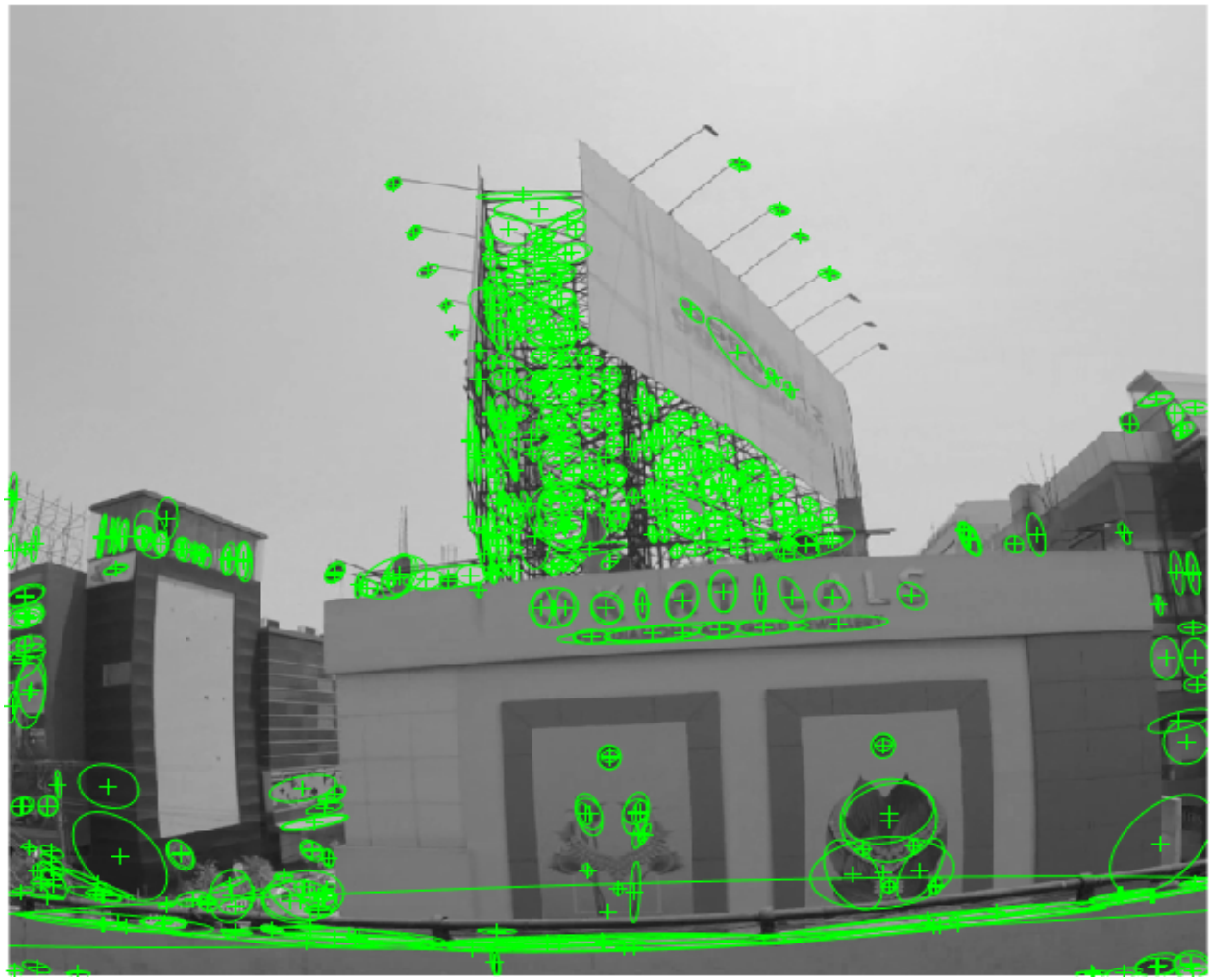
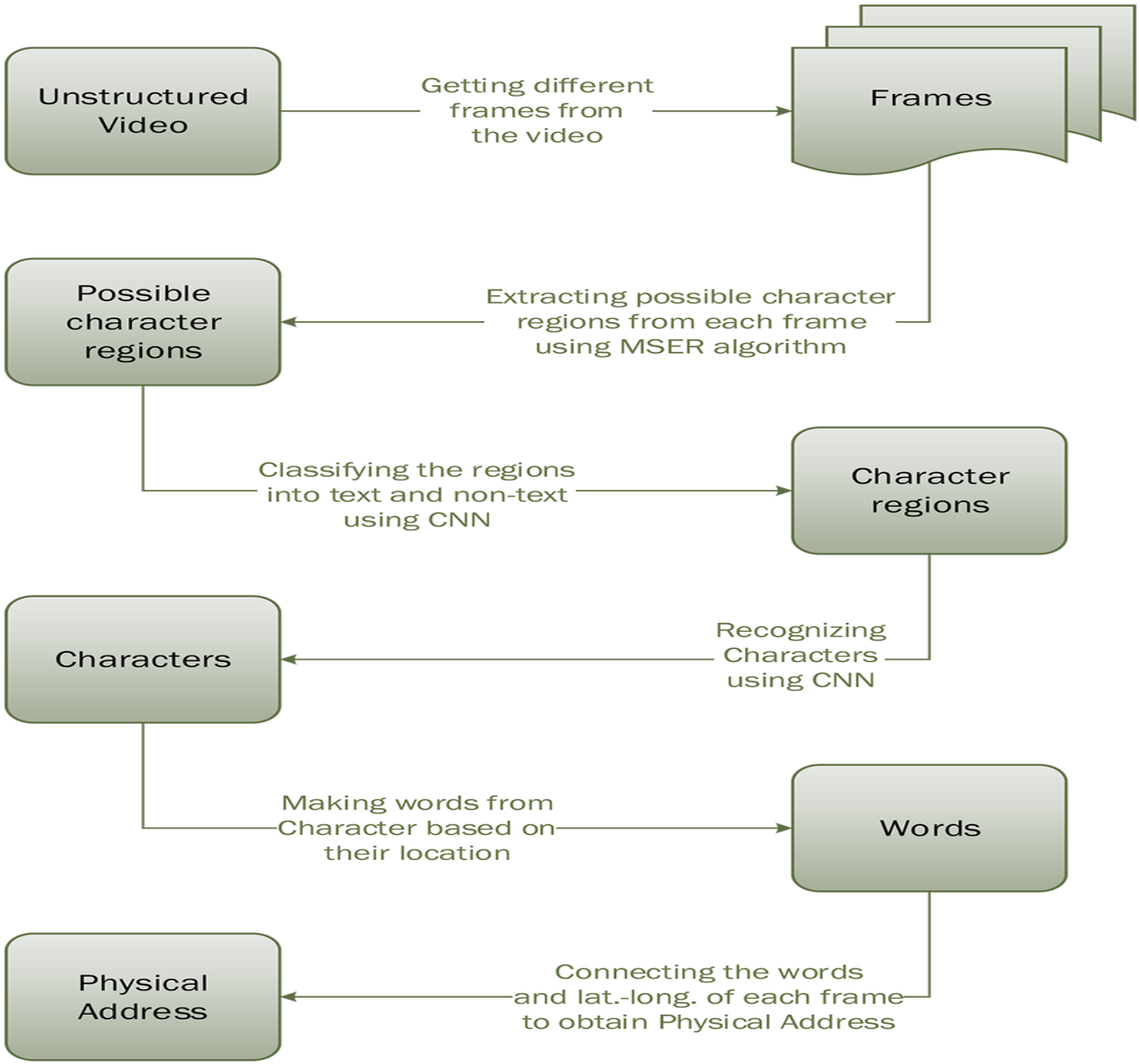
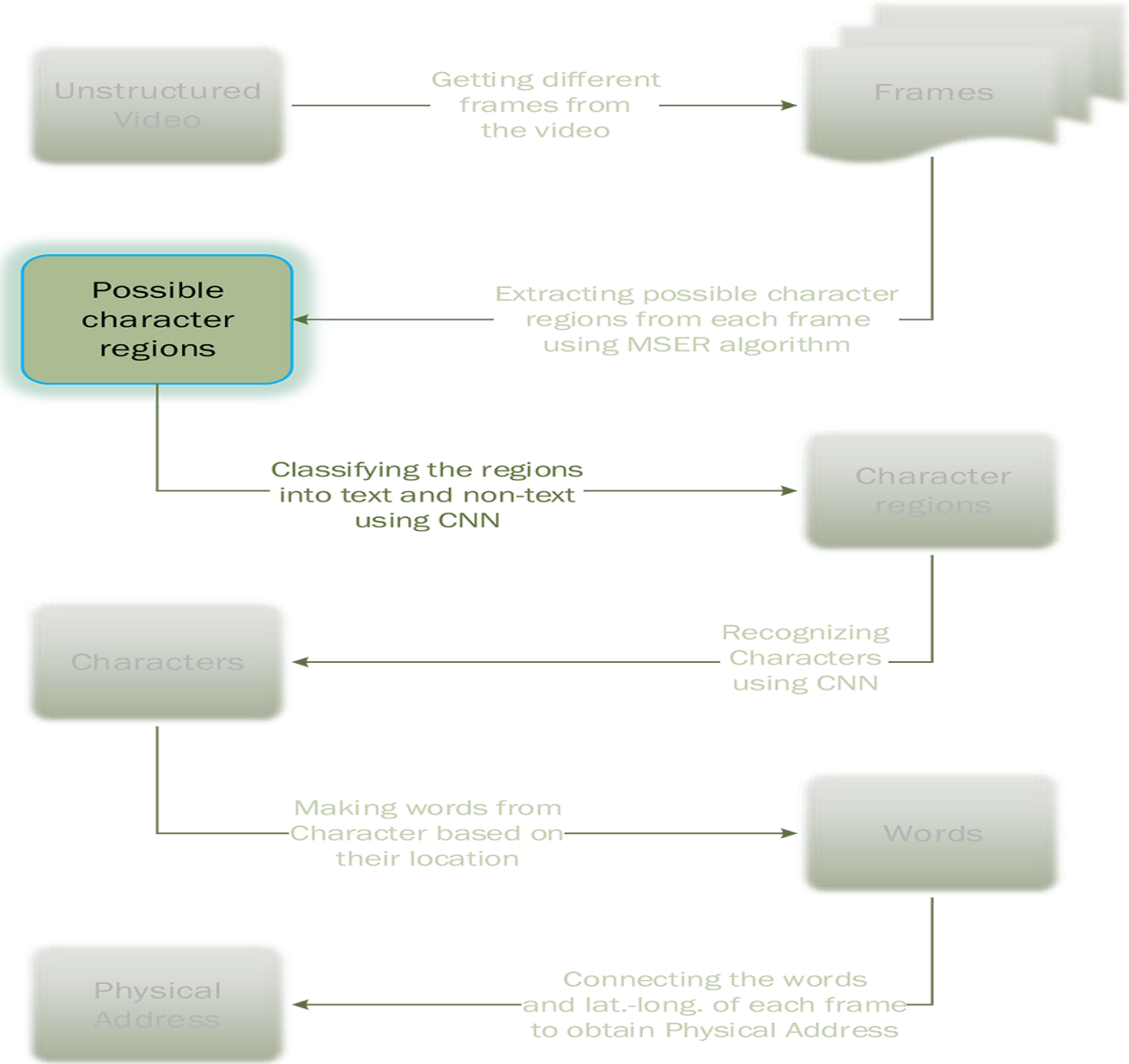


Figure 7: MSERs initially identified on video frame

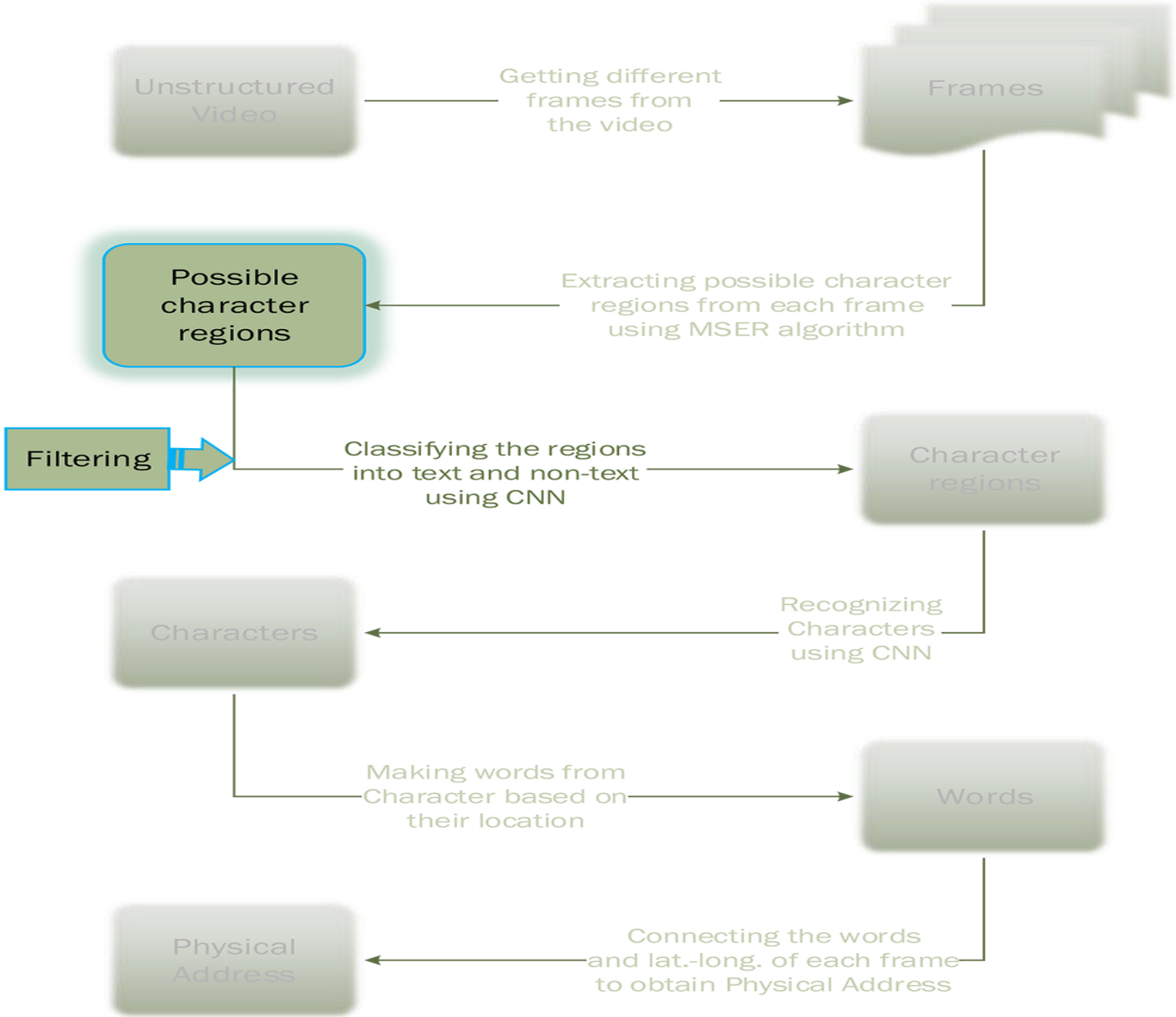
Work-flow


Unstructured Video
Getting different frames from the video
Frames
Possible character regions
Extracting possible character regions from each frame using MSER algorithm
Classifying the regions into text and non-text using CNN
Character regions
Characters
Recognizing Characters using CNN
Making words from Character based on their location
Words
Physical Address
Connecting the words and lat.-long. of each frame to obtain Physical Address

Work-flow


Unstructured Video
Getting different frames from the video
Frames
Possible character regions
Extracting possible character regions from each frame using MSER algorithm
Classifying the regions into text and non-text using CNN
Character regions
Characters
Recognizing Characters using CNN
Making words from Character based on their location
Words
Physical Address
Connecting the words and lat.-long. of each frame to obtain Physical Address

Work-flow


Unstructured Video
Getting different frames from the video
Frames
Possible character regions
Extracting possible character regions from each frame using MSER algorithm
Filtering
Classifying the regions into text and non-text using CNN
Character regions
Characters
Recognizing Characters using CNN
Making words from Character based on their location
Words
Physical Address
Connecting the words and lat.-long. of each frame to obtain Physical Address

MSERs - Initial Filtering

- There are overriding regions which represent same regions, this can be eliminated using the position of each regions.

MSERs - Initial Filtering

- There are overriding regions which represent same regions, this can be eliminated using the position of each regions.
- There are regions in which the width by height ratio is very high, these can also be removed by keeping a threshold on width by height ratio.

MSERs - Observations

- After initial filtering

Figure 8: Filtered MSERs on ICDAR 2015 image

MSERs - Observations

- After initial filtering

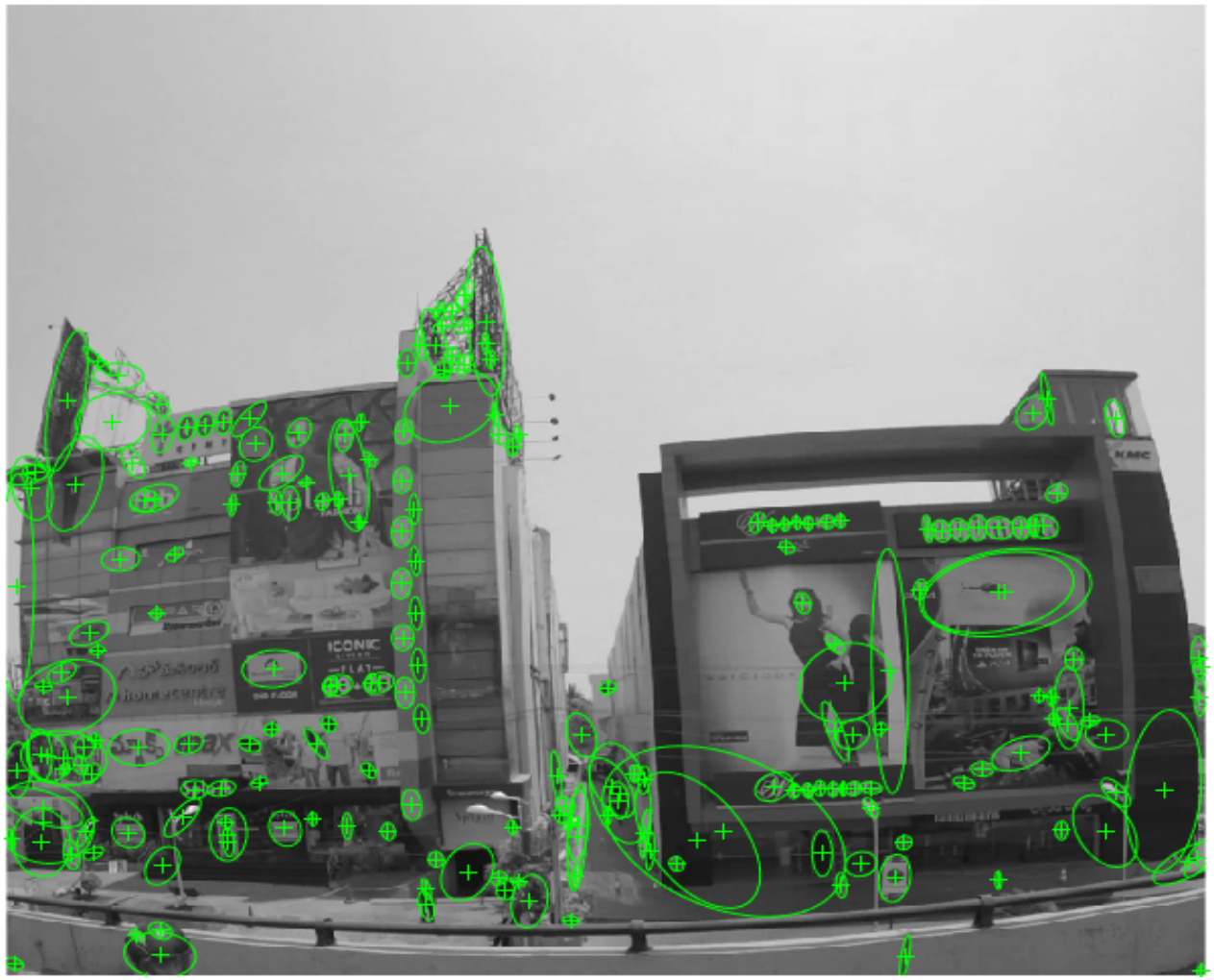


Figure 9: Filtered MSERs on video frame

MSERs - Observations

- After initial filtering

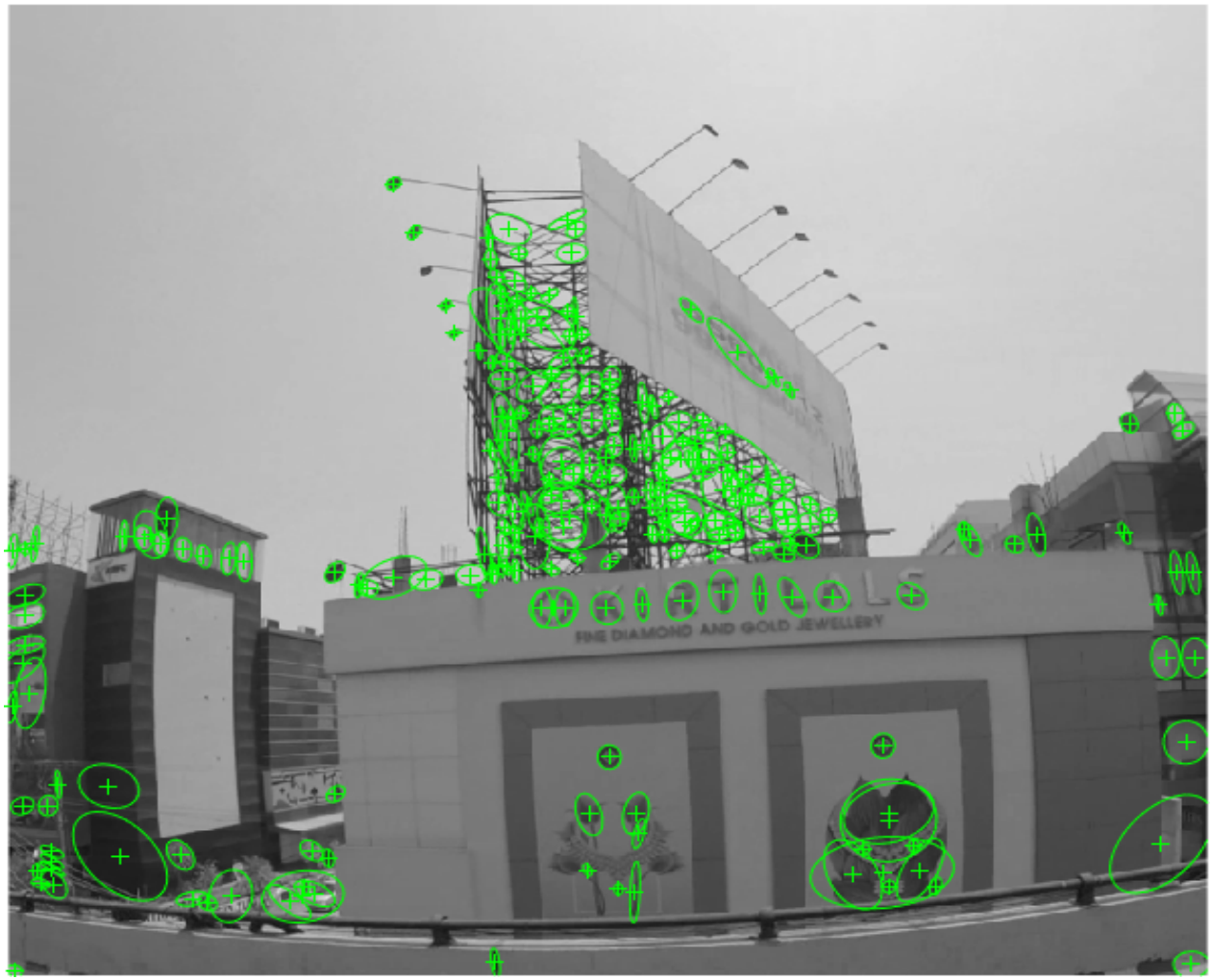


Figure 10: MSERs initially identified on video frame

Text and Non-Text Classification

- These filtered MSERs can be classified into text and non-text by using a suitable classifier.

Text and Non-Text Classification

- These filtered MSERs can be classified into text and non-text by using a suitable classifier.
- We used Convolutional Neural Network (CNN) [2, 5] for classification purpose with configurations:
  - Image input layer of size 24x24.

Text and Non-Text Classification

- These filtered MSERs can be classified into text and non-text by using a suitable classifier.
- We used Convolutional Neural Network (CNN) [2, 5] for classification purpose with configurations:
  - Image input layer of size 24x24.
  - Convolutional layer with 20 filters, each with a height & width of 5.

Text and Non-Text Classification

- These filtered MSERs can be classified into text and non-text by using a suitable classifier.
- We used Convolutional Neural Network (CNN) [2, 5] for classification purpose with configurations:
  - Image input layer of size 24x24.
  - Convolutional layer with 20 filters, each with a height & width of 5.
  - Relu Layer for thresholding

Text and Non-Text Classification

- These filtered MSERs can be classified into text and non-text by using a suitable classifier.
- We used Convolutional Neural Network (CNN) [2, 5] for classification purpose with configurations:
  - Image input layer of size 24x24.
  - Convolutional layer with 20 filters, each with a height & width of 5.
  - Relu Layer for thresholding
  - Max pooling layer with non-overlapping pooling regions, which down-samples by a factor of 2.

Text and Non-Text Classification

- These filtered MSERs can be classified into text and non-text by using a suitable classifier.
- We used Convolutional Neural Network (CNN) [2, 5] for classification purpose with configurations:
  - Image input layer of size 24x24.
  - Convolutional layer with 20 filters, each with a height & width of 5.
  - Relu Layer for thresholding
  - Max pooling layer with non-overlapping pooling regions, which down-samples by a factor of 2.
  - Fully connected layer with an output size of 2 and input size of 2000 (which is output size of preceding max pooling layer: 2000).

Text and Non-Text Classification

- These filtered MSERs can be classified into text and non-text by using a suitable classifier.
- We used Convolutional Neural Network (CNN) [2, 5] for classification purpose with configurations:
  - Image input layer of size 24x24.
  - Convolutional layer with 20 filters, each with a height & width of 5.
  - Relu Layer for thresholding
  - Max pooling layer with non-overlapping pooling regions, which down-samples by a factor of 2.
  - Fully connected layer with an output size of 2 and input size of 2000 (which is output size of preceding max pooling layer: 2000).
  - Soft-max layer.

Classification - Observations

- Text and Non-Text classification applied on simple images

Classification - Observations

- Text and Non-Text classification applied on simple images

Figure 11: Classified Text regions on ICDAR 2015 image

Classification - Observations

- Text and Non-Text classification applied on video frames

Classification - Observations

- Text and Non-Text classification applied on video frames

Figure 12: Classified Text regions on video frame

Classification - Observations

- Text and Non-Text classification applied on video frames

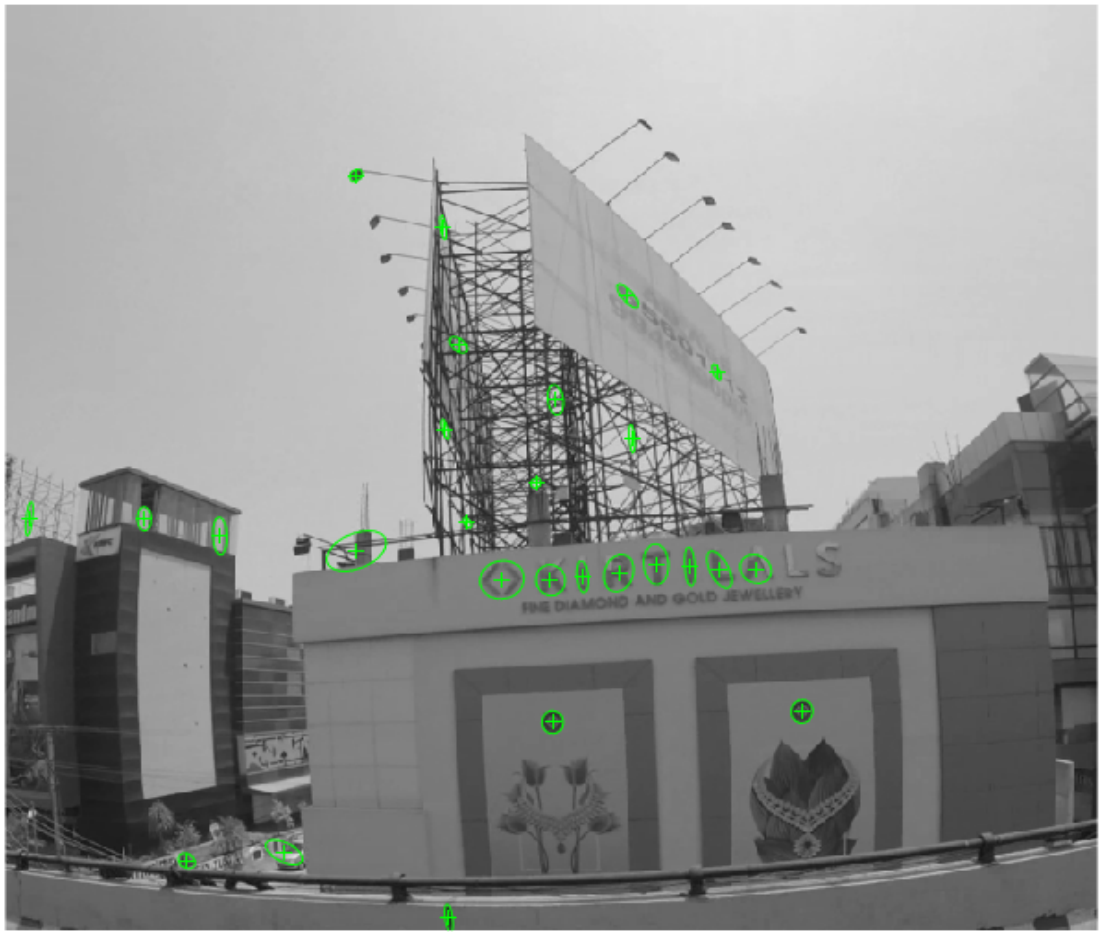
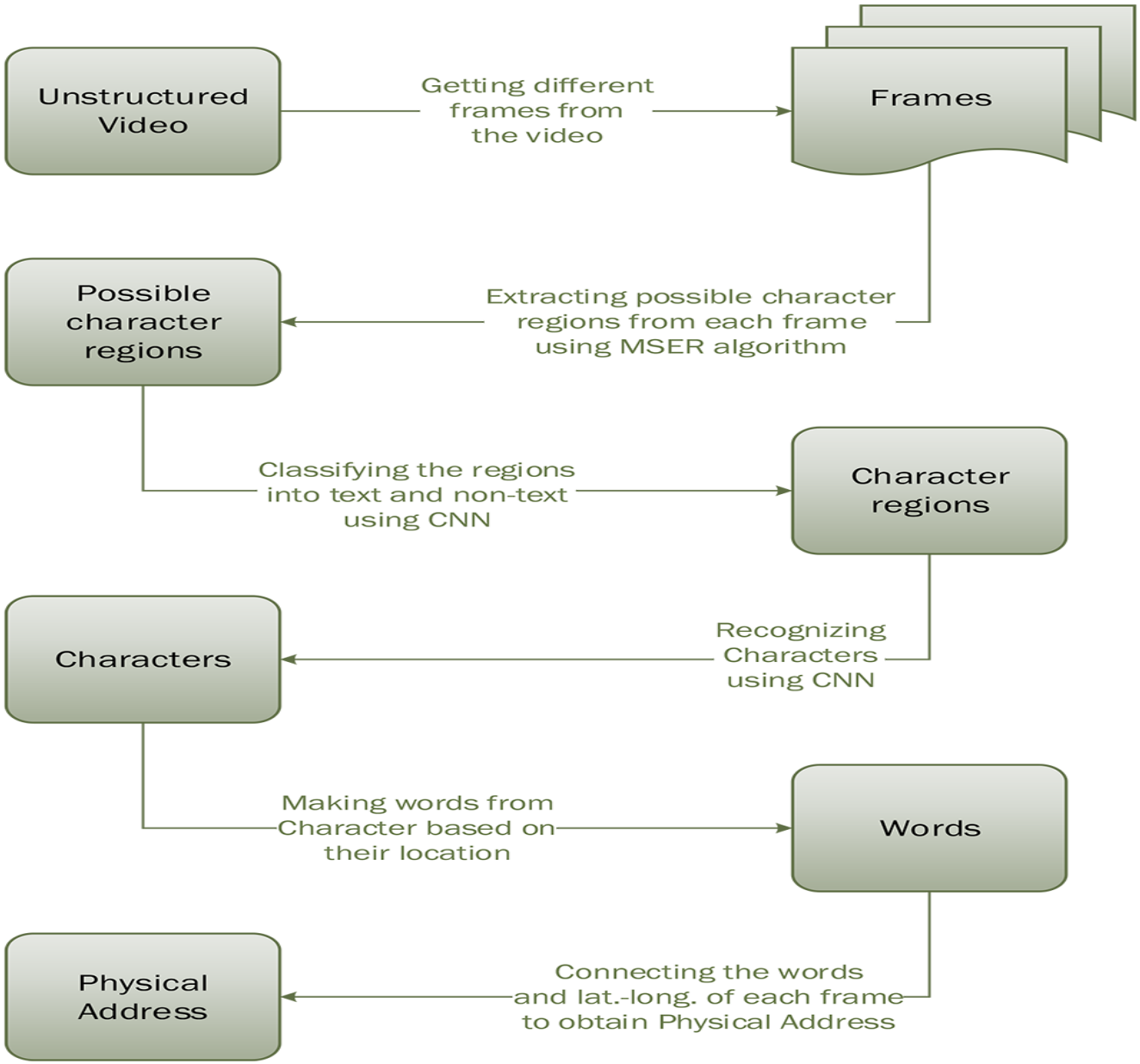
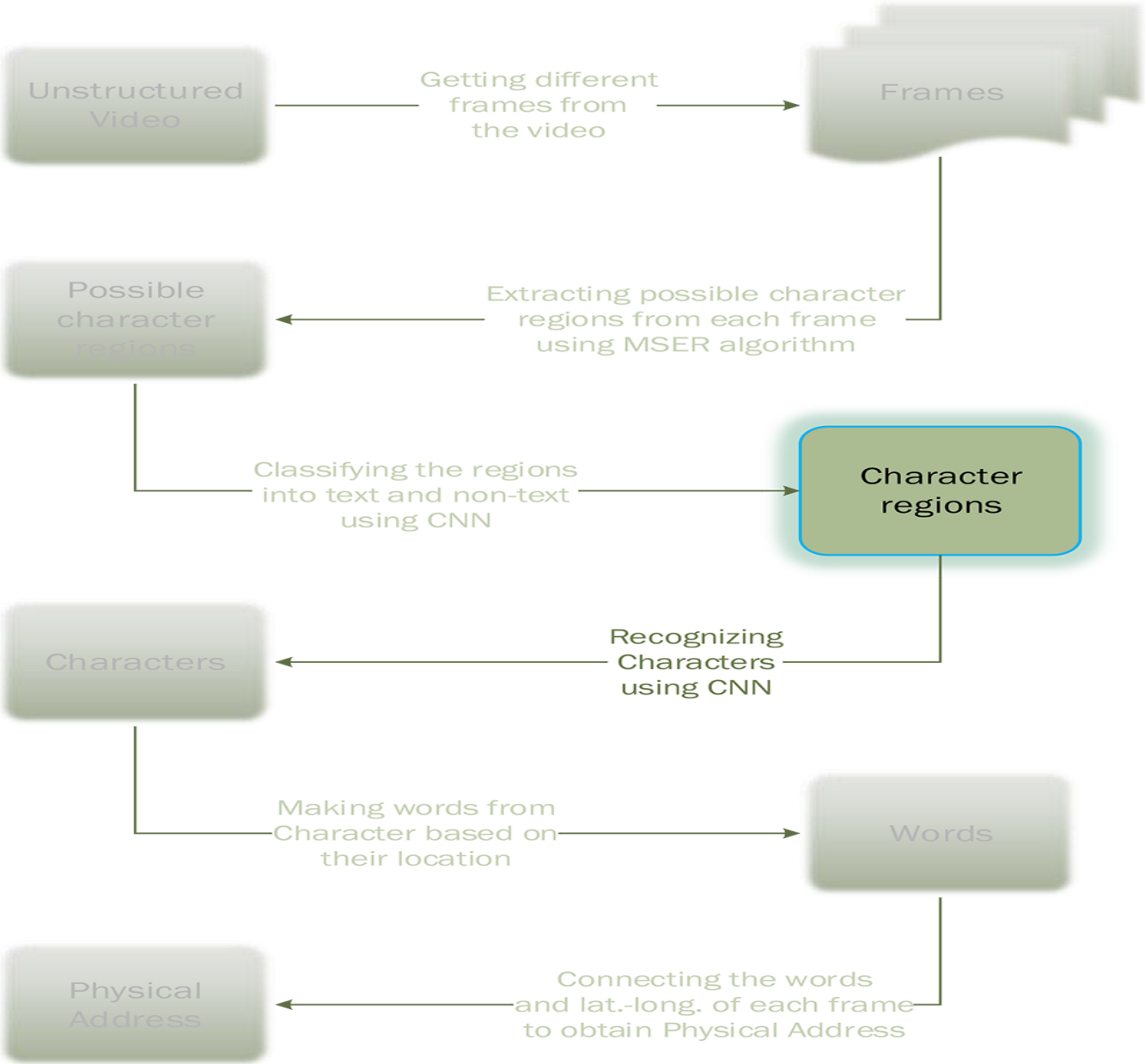


Figure 13: Classified Text regions on video frame

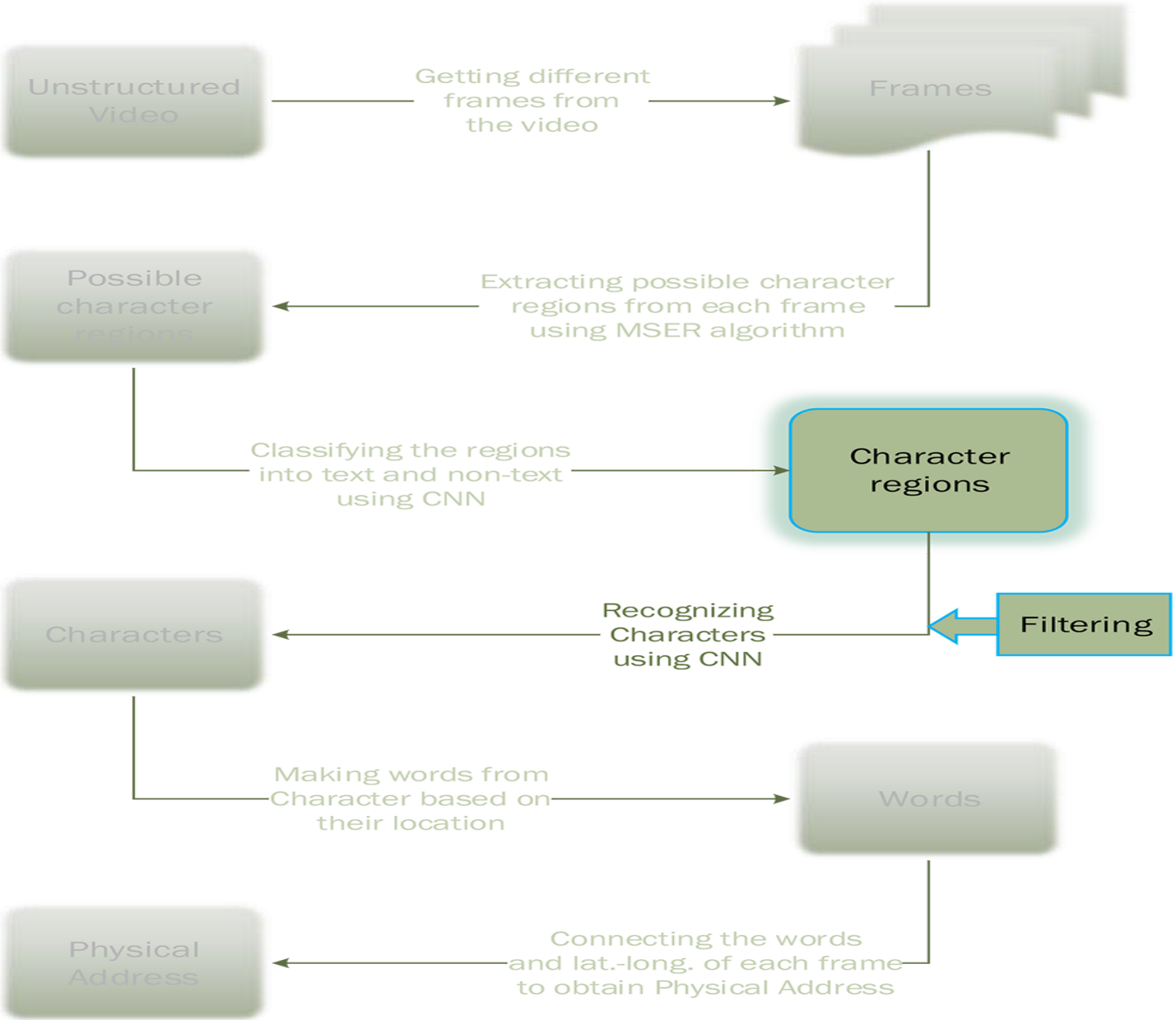
Work-flow


Unstructured Video
Getting different frames from the video
Frames
Possible character regions
Extracting possible character regions from each frame using MSER algorithm
Classifying the regions into text and non-text using CNN
Character regions
Characters
Recognizing Characters using CNN
Making words from Character based on their location
Words
Physical Address
Connecting the words and lat.-long. of each frame to obtain Physical Address

Work-flow


Unstructured Video
Getting different frames from the video
Frames
Possible character regions
Extracting possible character regions from each frame using MSER algorithm
Classifying the regions into text and non-text using CNN
Character regions
Characters
Recognizing Characters using CNN
Making words from Character based on their location
Words
Physical Address
Connecting the words and lat.-long. of each frame to obtain Physical Address

Work-flow


Unstructured Video
Getting different frames from the video
Frames
Possible character regions
Extracting possible character regions from each frame using MSER algorithm
Classifying the regions into text and non-text using CNN
Character regions
Characters
Recognizing Characters using CNN
Filtering
Making words from Character based on their location
Words
Physical Address
Connecting the words and lat.-long. of each frame to obtain Physical Address

Classification - Initial Filtering

- After Text and Non-Text classification, there are still some false positives.

Classification - Initial Filtering

- After Text and Non-Text classification, there are still some false positives.
- Regions classified as Text should occur with a uniform distance between the characters and at-least 2 together (we are assuming that there will not be any word in the image with only one character).

Classification - Initial Filtering

- After Text and Non-Text classification, there are still some false positives.
- Regions classified as Text should occur with a uniform distance between the characters and at-least 2 together (we are assuming that there will not be any word in the image with only one character).
- After applying this filter based on distance between the center of regions, we are left with just text regions.

Classification - Observations

- Filtering applied on classified Text regions

Classification - Observations

- Filtering applied on classified Text regions

Figure 14: ICDAR Images

Classification - Observations

- Filtering applied on classified Text regions

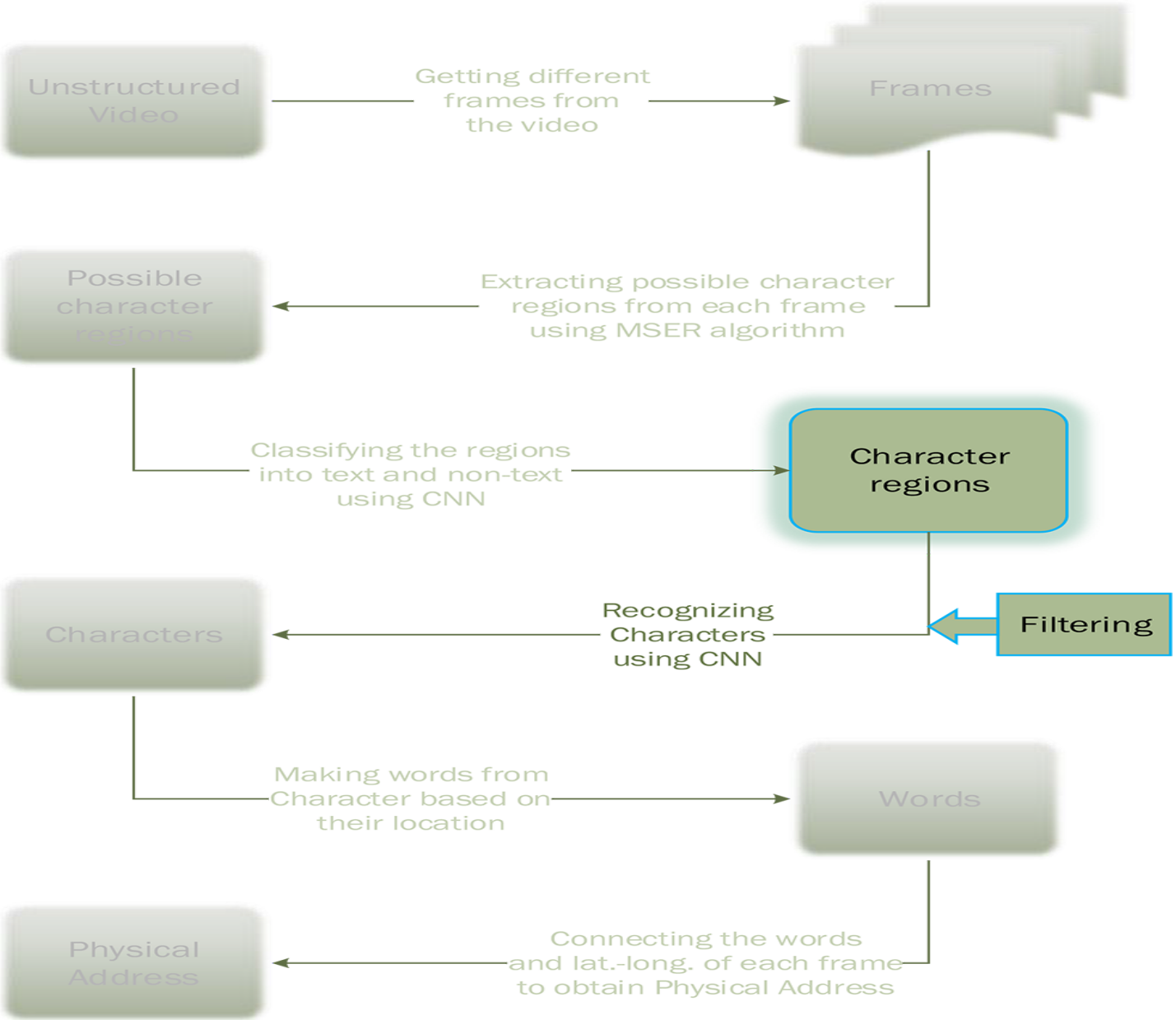
Figure 15: Video frames

Classification - Observations

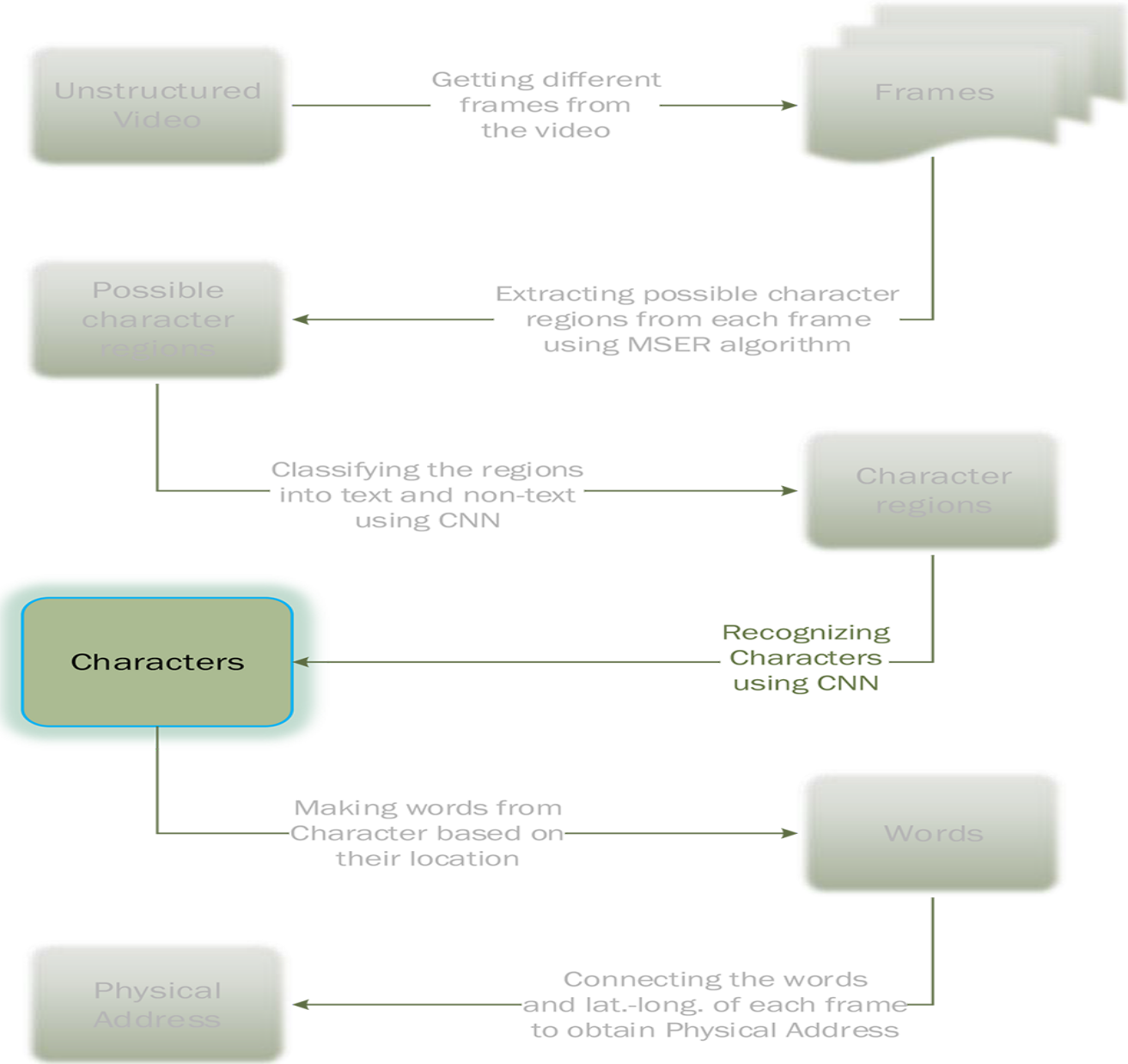
- Filtering applied on classified Text regions

Figure 16: Video frames

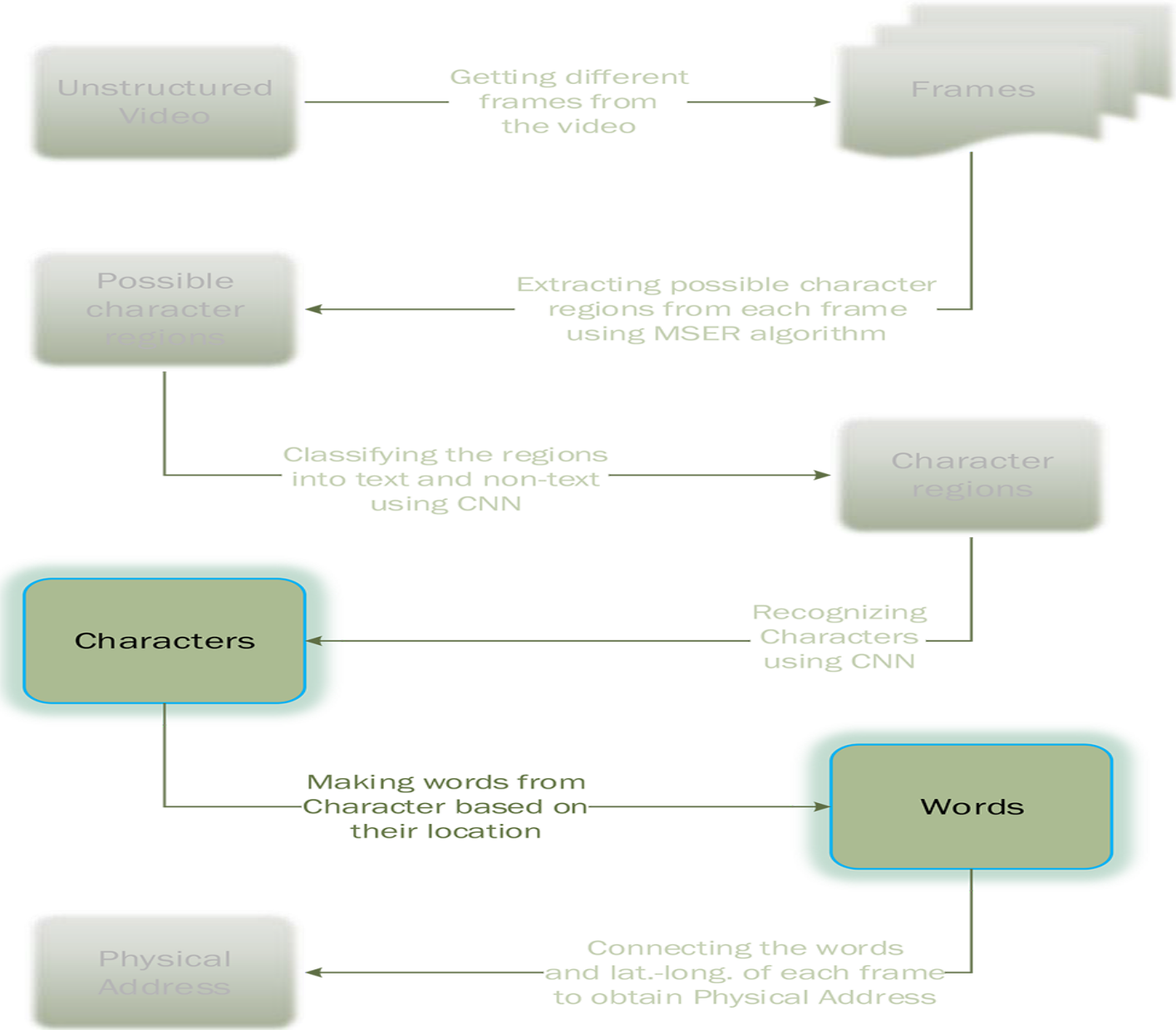
Work-flow


Unstructured Video
Getting different frames from the video
Frames
Possible character regions
Extracting possible character regions from each frame using MSER algorithm
Classifying the regions into text and non-text using CNN
Character regions
Characters
Recognizing Characters using CNN
Filtering
Making words from Character based on their location
Words
Physical Address
Connecting the words and lat.-long. of each frame to obtain Physical Address

Work-flow


Unstructured Video
Getting different frames from the video
Frames
Possible character regions
Extracting possible character regions from each frame using MSER algorithm
Classifying the regions into text and non-text using CNN
Character regions
Characters
Recognizing Characters using CNN
Making words from Character based on their location
Words
Physical Address
Connecting the words and lat.-long. of each frame to obtain Physical Address

Work-flow


Unstructured Video
Getting different frames from the video
Frames
Possible character regions
Extracting possible character regions from each frame using MSER algorithm
Classifying the regions into text and non-text using CNN
Character regions
Characters
Recognizing Characters using CNN
Making words from Character based on their location
Words
Physical Address
Connecting the words and lat.-long. of each frame to obtain Physical Address

Text Recognition - Observations

- Recognized texts on simple images

Text Recognition - Observations

- Recognized texts on simple images

Figure 17: ICDAR Images

Text Recognition - Observations

- Recognized texts on video frames

Text Recognition - Observations

- Recognized texts on video frames

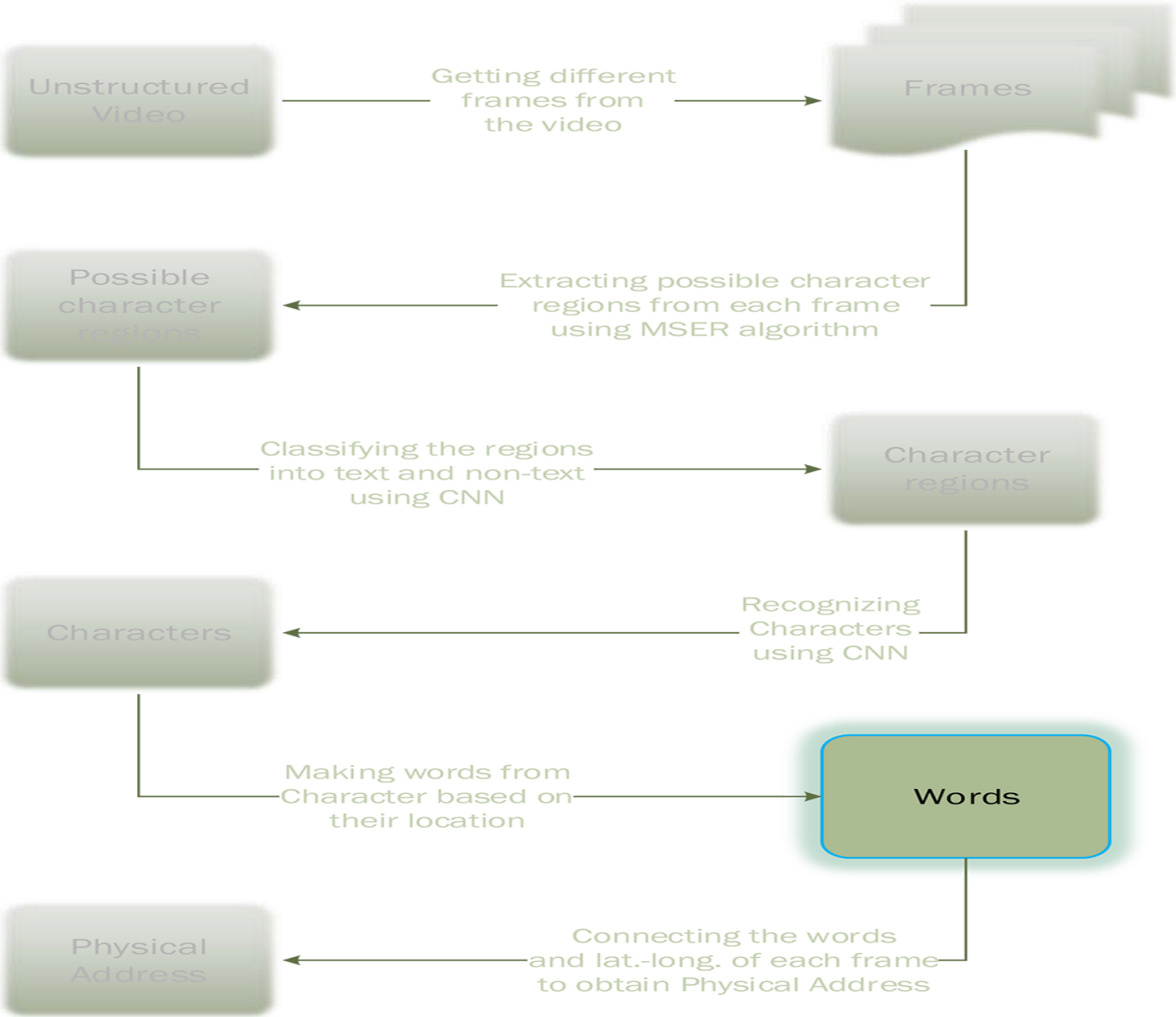
Figure 18: Video frames

Text Recognition - Observations

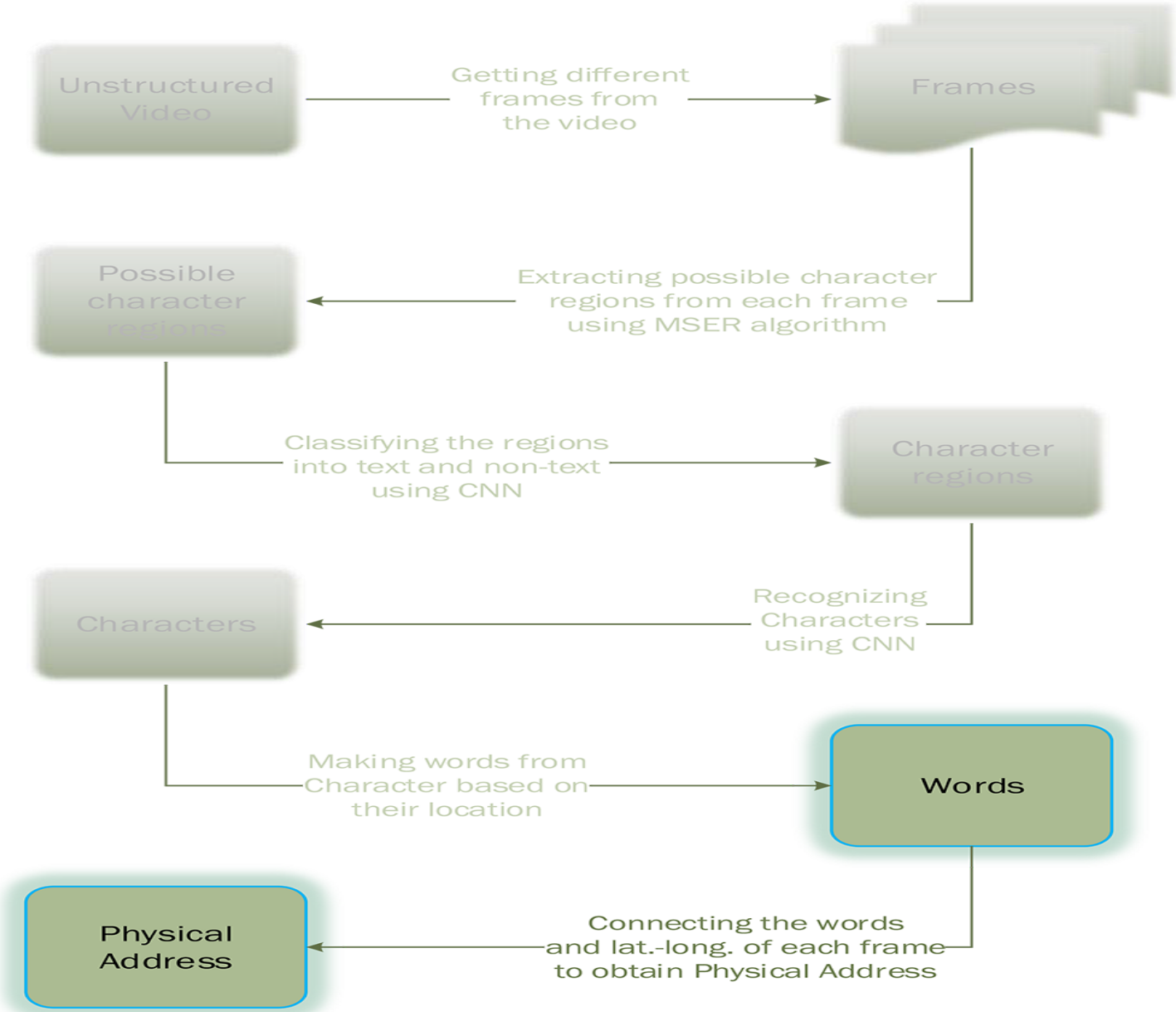
- Recognized texts on video frames

Figure 19: Video frames

Work-flow


Unstructured Video
Getting different frames from the video
Frames
Possible character regions
Extracting possible character regions from each frame using MSER algorithm
Classifying the regions into text and non-text using CNN
Character regions
Characters
Recognizing Characters using CNN
Making words from Character based on their location
Words
Physical Address
Connecting the words and lat.-long. of each frame to obtain Physical Address

Work-flow


Unstructured Video
Getting different frames from the video
Frames
Possible character regions
Extracting possible character regions from each frame using MSER algorithm
Classifying the regions into text and non-text using CNN
Character regions
Characters
Recognizing Characters using CNN
Making words from Character based on their location
Words
Physical Address
Connecting the words and lat.-long. of each frame to obtain Physical Address

Results and Comparison

- Criteria for quantification:

Results and Comparison

- Criteria for quantification:
  - All characters in the image which are atleast two in length and with reasonable clarity and luminosity should be recognized.

Results and Comparison

- Criteria for quantification:
  - All characters in the image which are atleast two in length and with reasonable clarity and luminosity should be recognized.
- On simple images:
  - Greater than 99.5% of character regions are identified.

Results and Comparison

- Criteria for quantification:
  - All characters in the image which are atleast two in length and with reasonable clarity and luminosity should be recognized.
- On simple images:
  - Greater than 99.5% of character regions are identified.
  - Greater than 99% of characters were recognized.

Results and Comparison

- Criteria for quantification:
  - All characters in the image which are atleast two in length and with reasonable clarity and luminosity should be recognized.
- On simple images:
  - Greater than 99.5% of character regions are identified.
  - Greater than 99% of characters were recognized.
- On video frames:
  - 40% to 50% of character regions are identified.

Results and Comparison

- Criteria for quantification:
  - All characters in the image which are atleast two in length and with reasonable clarity and luminosity should be recognized.
- On simple images:
  - Greater than 99.5% of character regions are identified.
  - Greater than 99% of characters were recognized.
- On video frames:
  - 40% to 50% of character regions are identified.
  - Greater than 99% of characters were recognized.

Conclusions and Recommendations

- Worked well for ICDAR Dataset, but average for video frames.

Conclusions and Recommendations

- Worked well for ICDAR Dataset, but average for video frames.
- Possible Reason:
  - Images not well focused and are with very less quality.

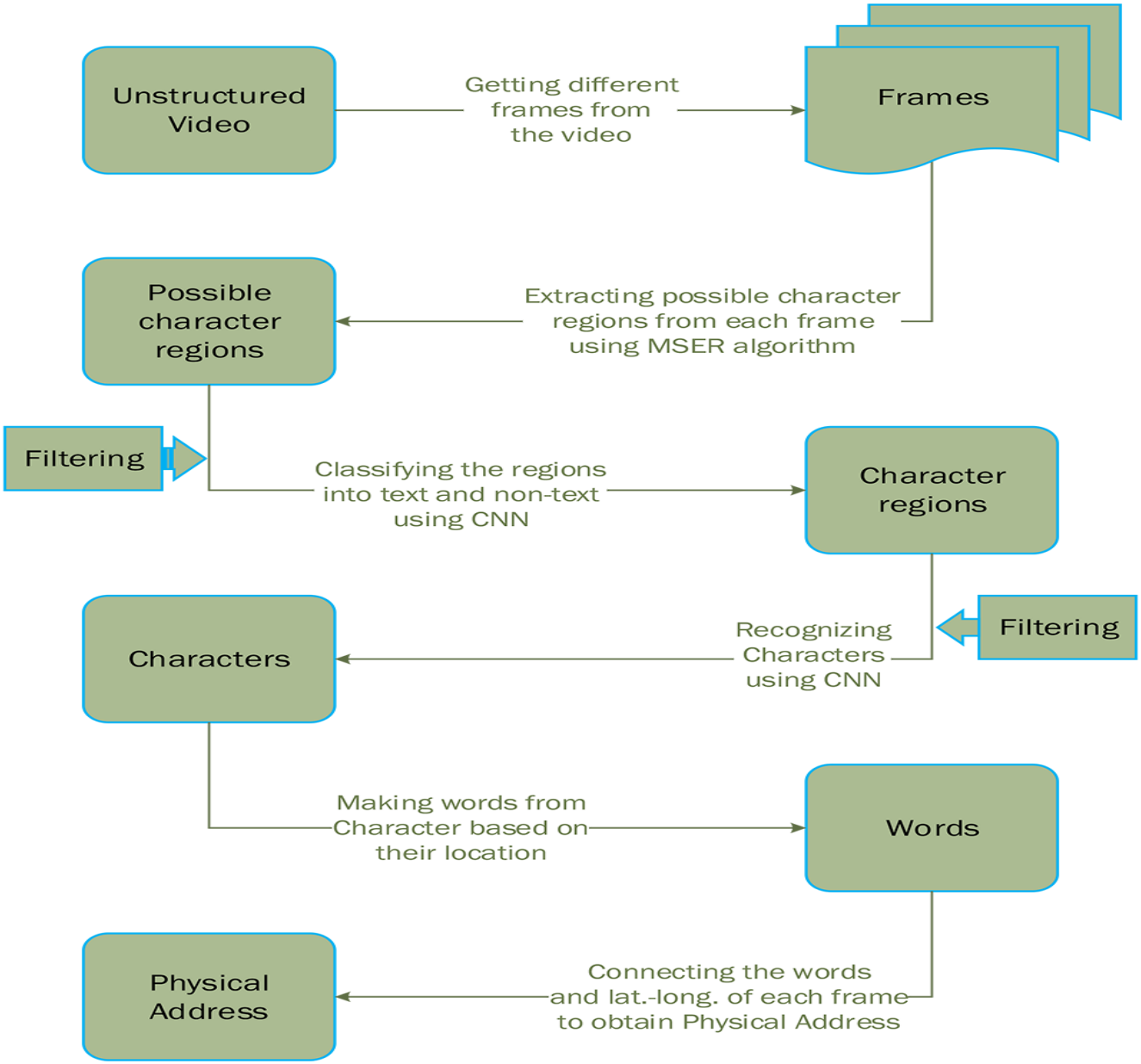
Conclusions and Recommendations

- Worked well for ICDAR Dataset, but average for video frames.
- Possible Reason:
  - Images not well focused and are with very less quality.
- Recommendations:
  - Use **extremal regions** instead of **MSERs** (Computationally very expensive).

Conclusions and Recommendations

- Worked well for ICDAR Dataset, but average for video frames.
- Possible Reason:
  - Images not well focused and are with very less quality.
- Recommendations:
  - Use **extremal regions** instead of **MSERs** (Computationally very expensive).
  - Use of well tuned classifier (CNN configuration tuned and trained for this specific dataset).

Summary


Unstructured Video
Getting different frames from the video
Frames
Possible character regions
Extracting possible character regions from each frame using MSER algorithm
Filtering
Classifying the regions into text and non-text using CNN
Character regions
Characters
Recognizing Characters using CNN
Filtering
Making words from Character based on their location
Words
Physical Address
Connecting the words and lat.-long. of each frame to obtain Physical Address

References

[1] ICDAR 2015. "Incidental Scene Text Dataset". In: *International Conference on Document Analysis and Recognition* (2015).

[2] I. Sutskever A. Krizhevsky and G. E. Hinton. "ImageNet classification with deep convolutional neural network". In: *Proc. Conference on Neural Information Processing Systems* (2012), pp. 1097–1105.

[3] Larry Brown. *Using cuDNN depth neural network*. 2014. URL: https://blogs.nvidia.com.tw/2014/09/accelerate-machine-learning-cudnn-deep-neural-network-library/ (visited on 03/30/2017).

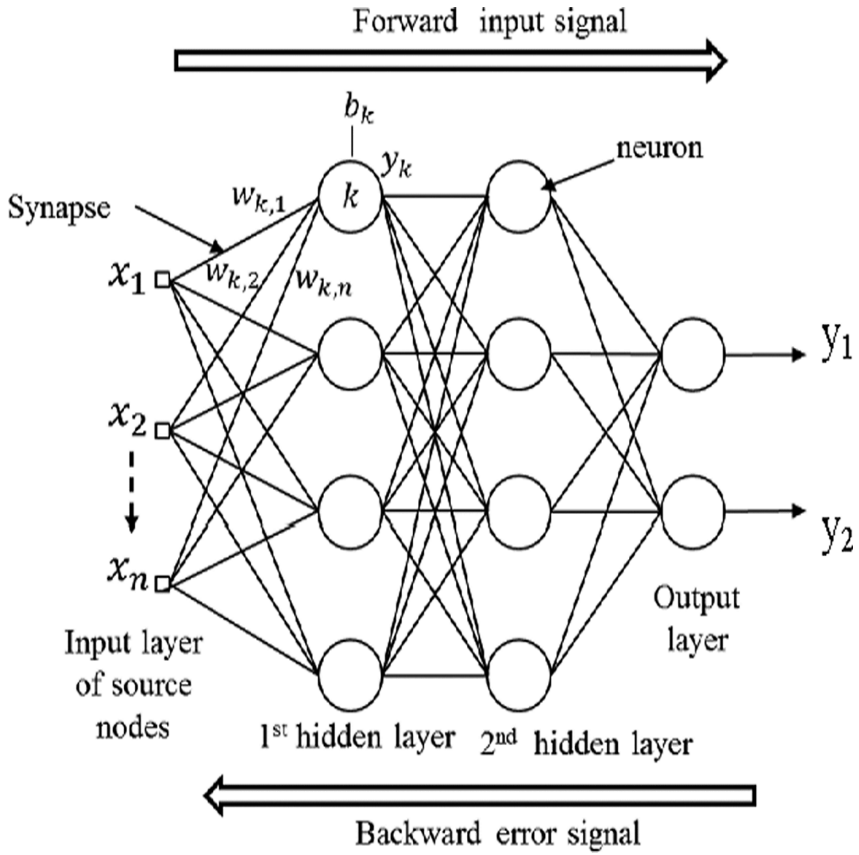
[4] M. Urban J. Matas O. Chum and T. Pajdla. "Robust wide baseline stereo from maximally stable extremal regions". In: *Proc. British Machine Vision Conference* (2002), pp. 384–396.

[5] Y. Boureau K. Gregor M. Mathieu K. Kavukcuoglu P. Sermanet and Y. LeCun. "ImageNet classification with deep convolutional neural network". In: *Proc. Conference on Neural Information Processing Systems* (2012), pp. 1097–1105.

Discussion

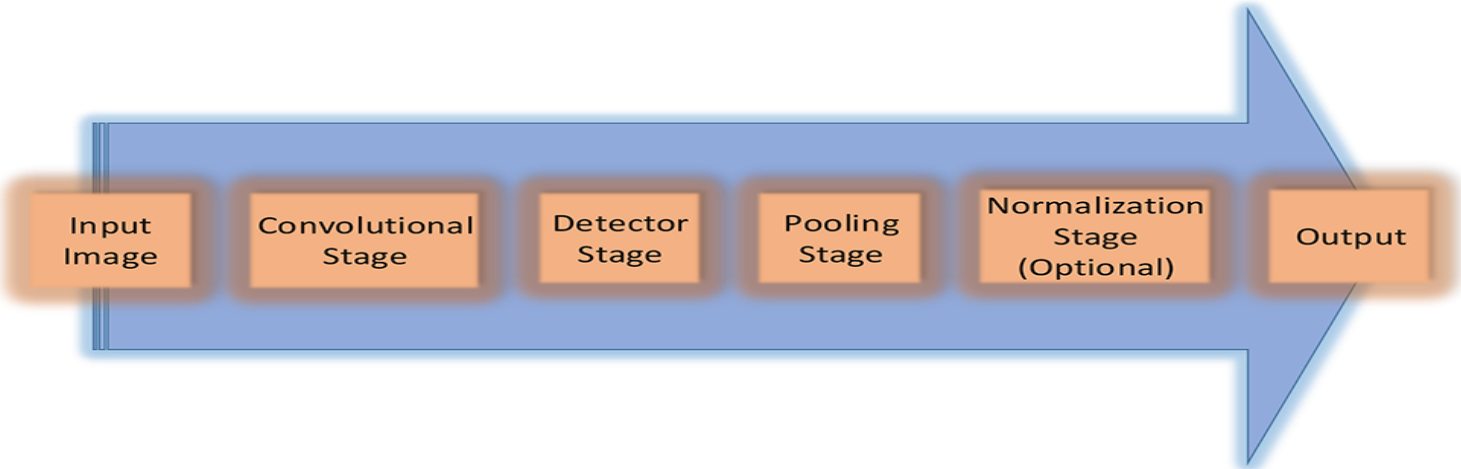
Thank you

Convolutional Neural Network



- CNN is neural networks that use convolution in place of general matrix multiplication in at least one of their layers.

Convolutional Neural Network



[2] Larry Brown. *Using cuDNN depth neural network*. 2014. URL: https://blogs.nvidia.com.tw/2014/09/accelerate-machine-learning-cudnn-deep-neural-network-library/ (visited on 03/30/2017).

Convolutional Neural Network

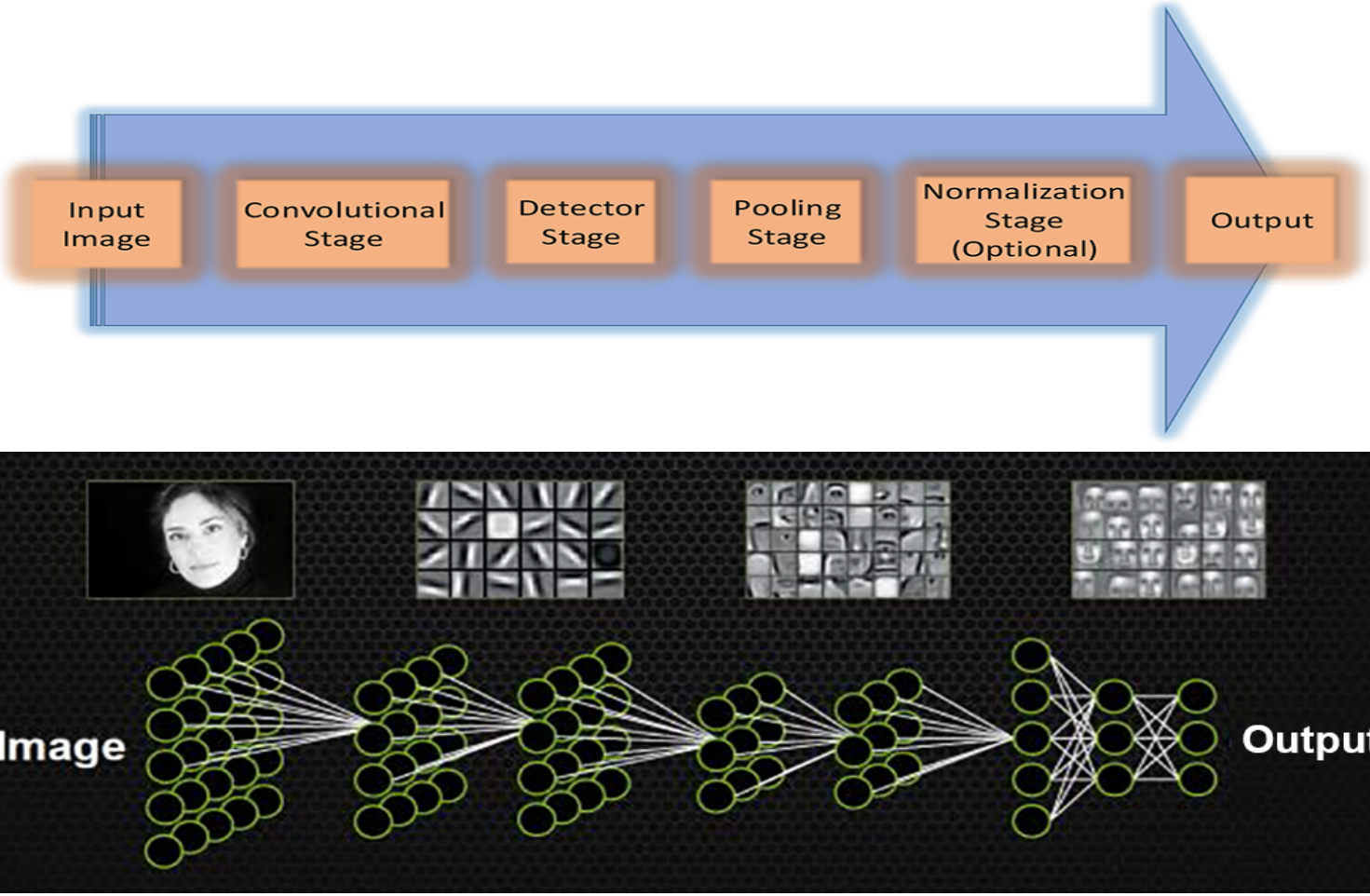


Figure 20: CNN work-flow[2]

back

[2] Larry Brown. *Using cuDNN depth neural network*. 2014. URL: https://blogs.nvidia.com.tw/2014/09/accelerate-machine-learning-cudnn-deep-neural-network-library/ (visited on 03/30/2017).