Computer Vision: Technical Report-Project 1

- ***Filter Selection***

x-1,y+1	x,y+1	x+1,y+1
x-1,y	x,y	x+1,y
x-1,y-1	x,y-1	x+1,y-1

Image 1. Schematic Implementation of median filter

```
def median(imggr): #manual implementation of median filter
    imgf=imggr
    for x in range(0,(imggr.shape[0]-1)):
        for y in range(0,(imggr.shape[1]-1)):
            window = [imggr[0, 0]] * 9
            window[0] = imggr[x-1,y-1]
            window[1] = imggr[x,y-1]
            window[2] = imggr[x+1,y-1]
            window[3] = imggr[x-1,y]
            window[4] = imggr[x,y]
            window[5] = imggr[x+1,y]
            window[6] = imggr[x-1,y+1]
            window[7] = imggr[x,y+1]
            window[8] = imggr[x+1,y+1]
            window.sort()
            imgf[x,y]=window[4]
    return imgf
```

Image 2. Python implementation of median filter

The filter selected and implemented was a non-linear median filter. The selection was made based on the fact that this kind of filter is the optimal choice when dealing with salt & pepper noise. Because the filter uses the median value of all the neighborhood(window) values around the pixel that is of interest and replaces its value, a uniformed result is achieved. Two for-loops have been implemented, with the first one running on the first dimension and the second one on the second dimension. In more details, at first, a window with dimensions 3x3 is initialized with values of the first pixel, and then each pixel in the original image is replaced with the median value of the filter. However, the drawback of this implementation filter is that due

to the two for loops, there is an increased runtime. Even so, I believe that due to the quality of the end result, the increased runtime is worth it. Below are the median filtering results for the images given, compared with the original.

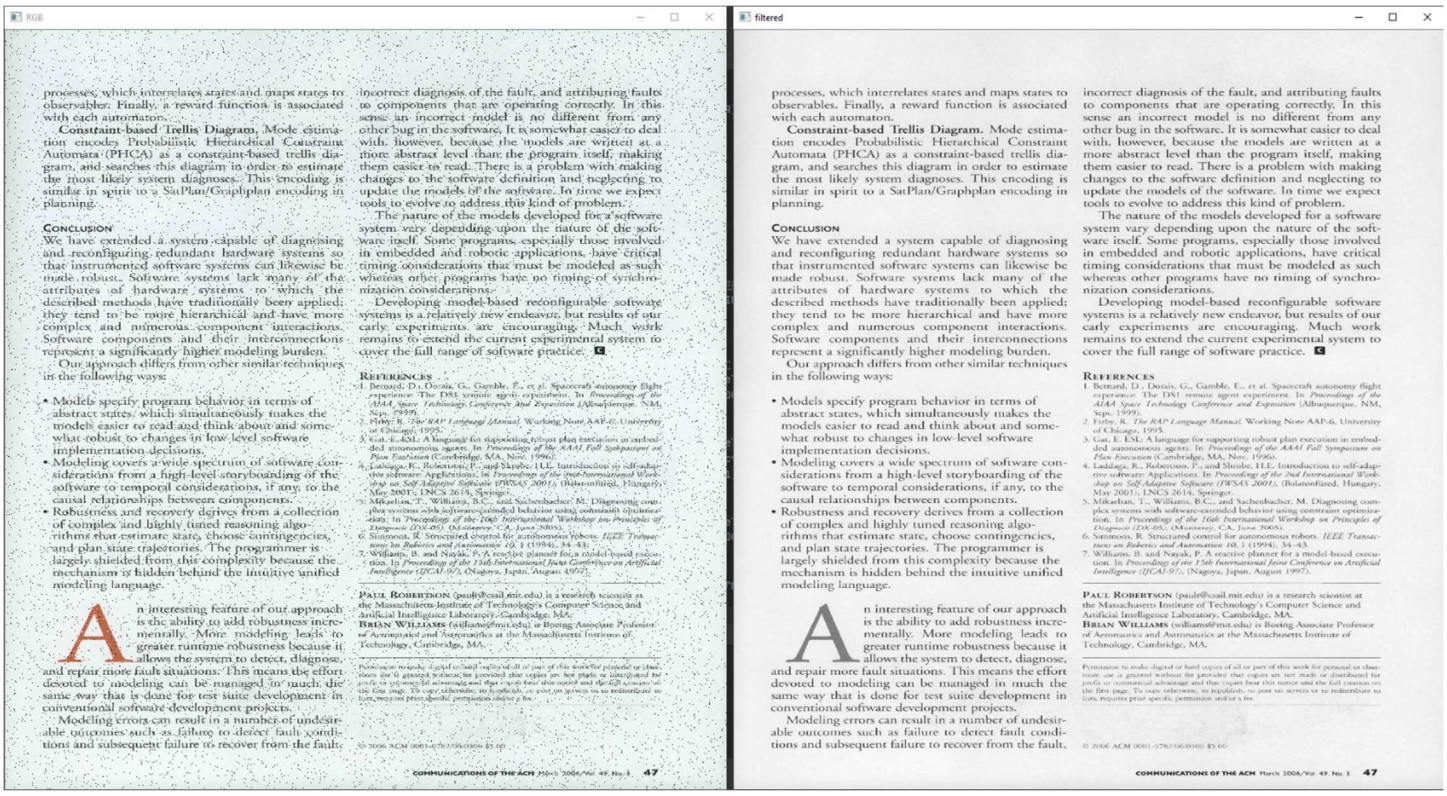


Image 3. Before filtering(left) and after(right) for image 2_original.png

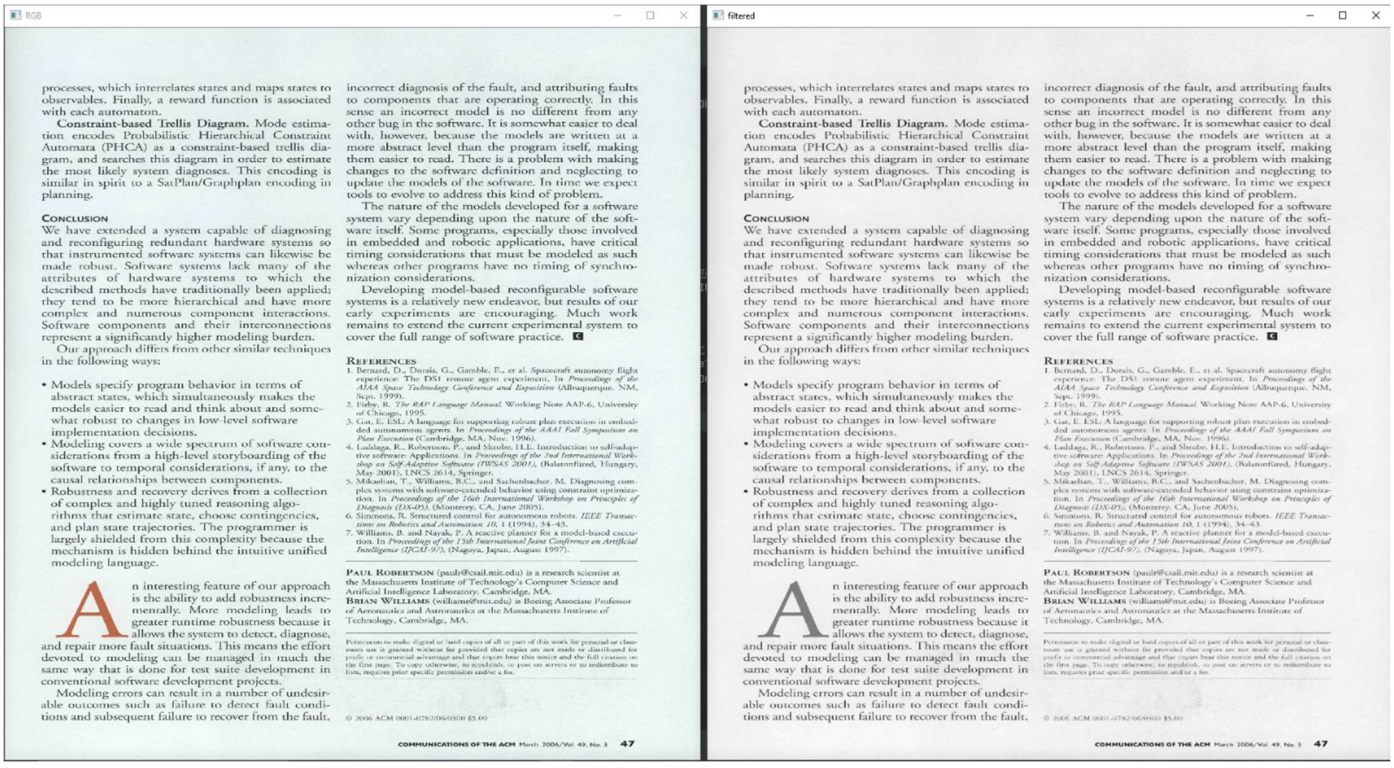


Image 4. Before filtering(left) and after(right) for image 2_noise.png

Image 5. Before filtering(left) and after(right) for image 3_original.png

Image 6. Before filtering(left) and after(right) for image 3_noise.png

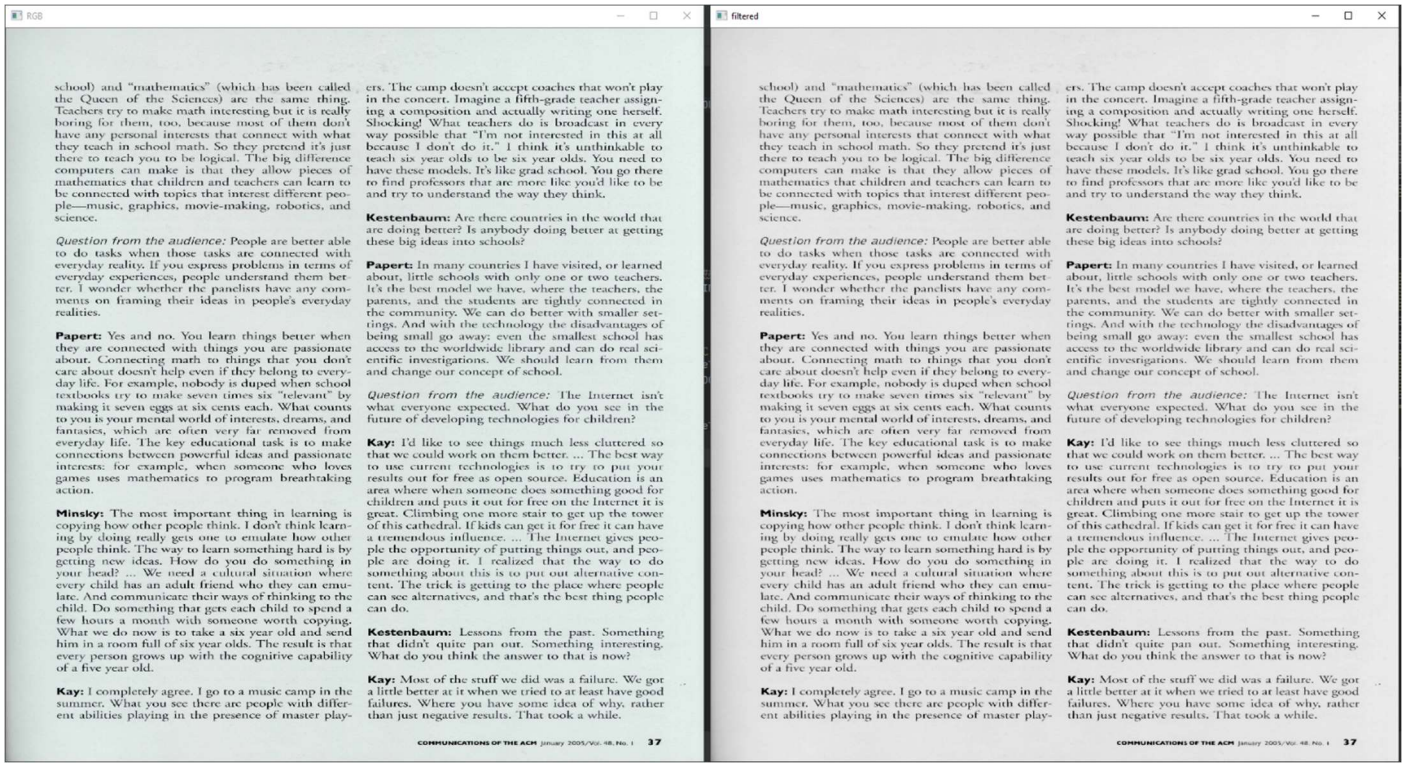


Image 7. Before filtering(left) and after(right) for image 4_original.png

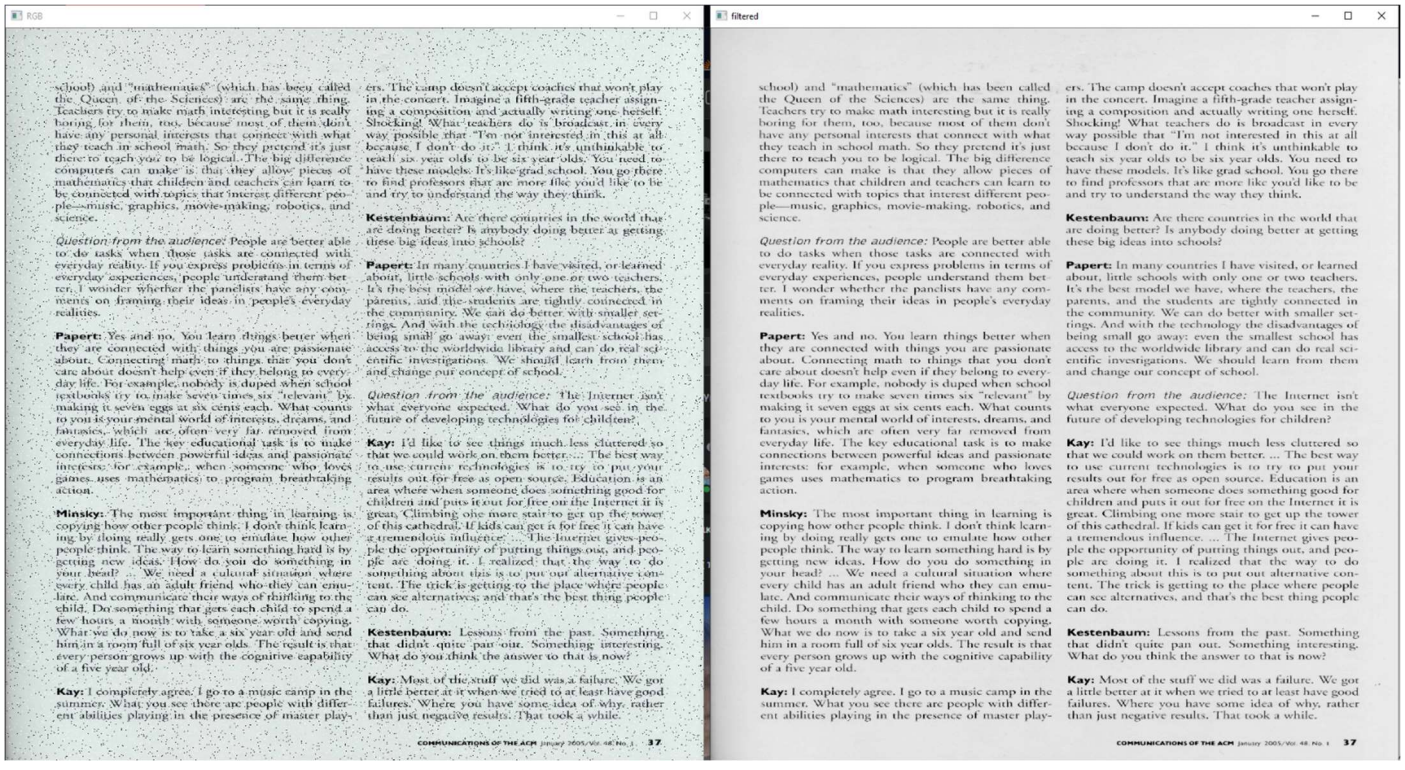


Image 8. Before filtering(left) and after(right) for image 4_noise.png

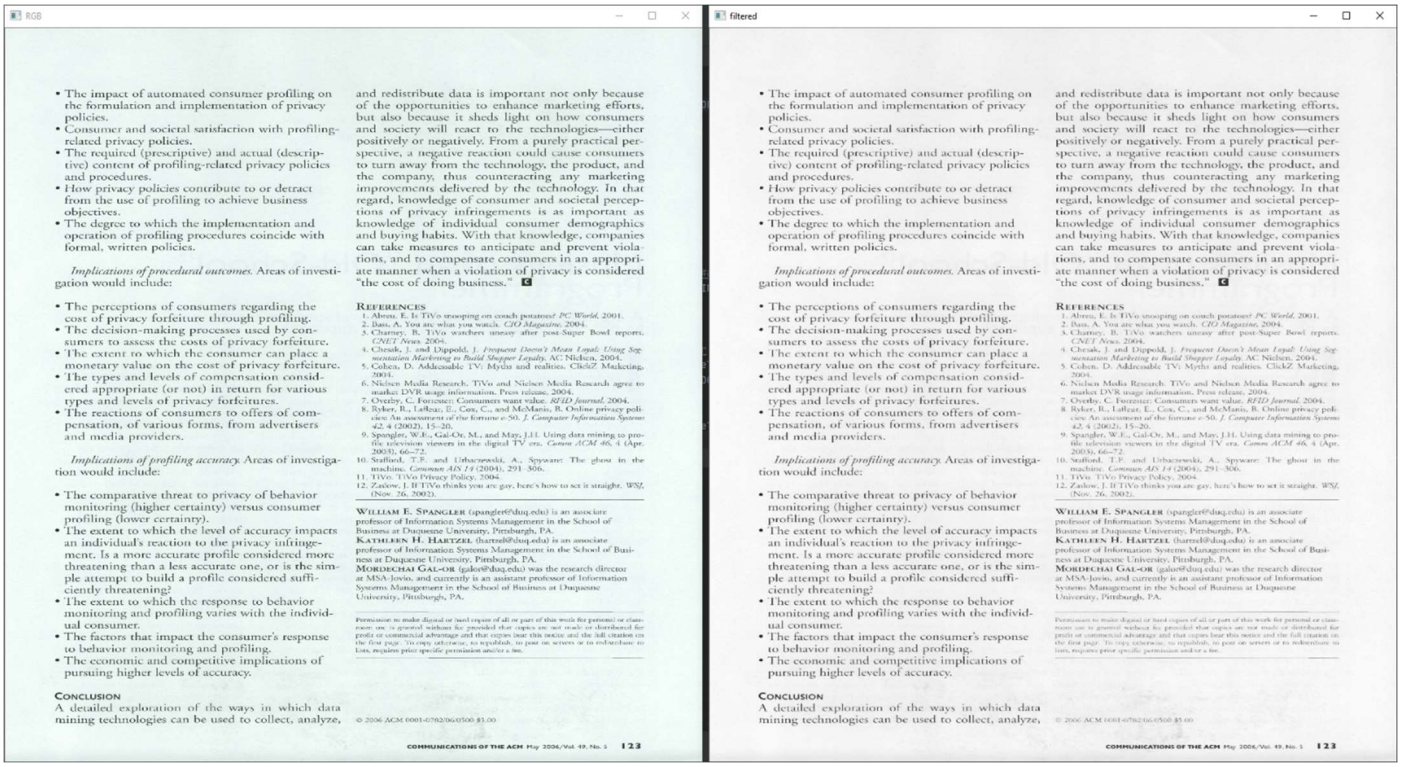


Image 9. Before filtering(left) and after(right) for image 5_original.png

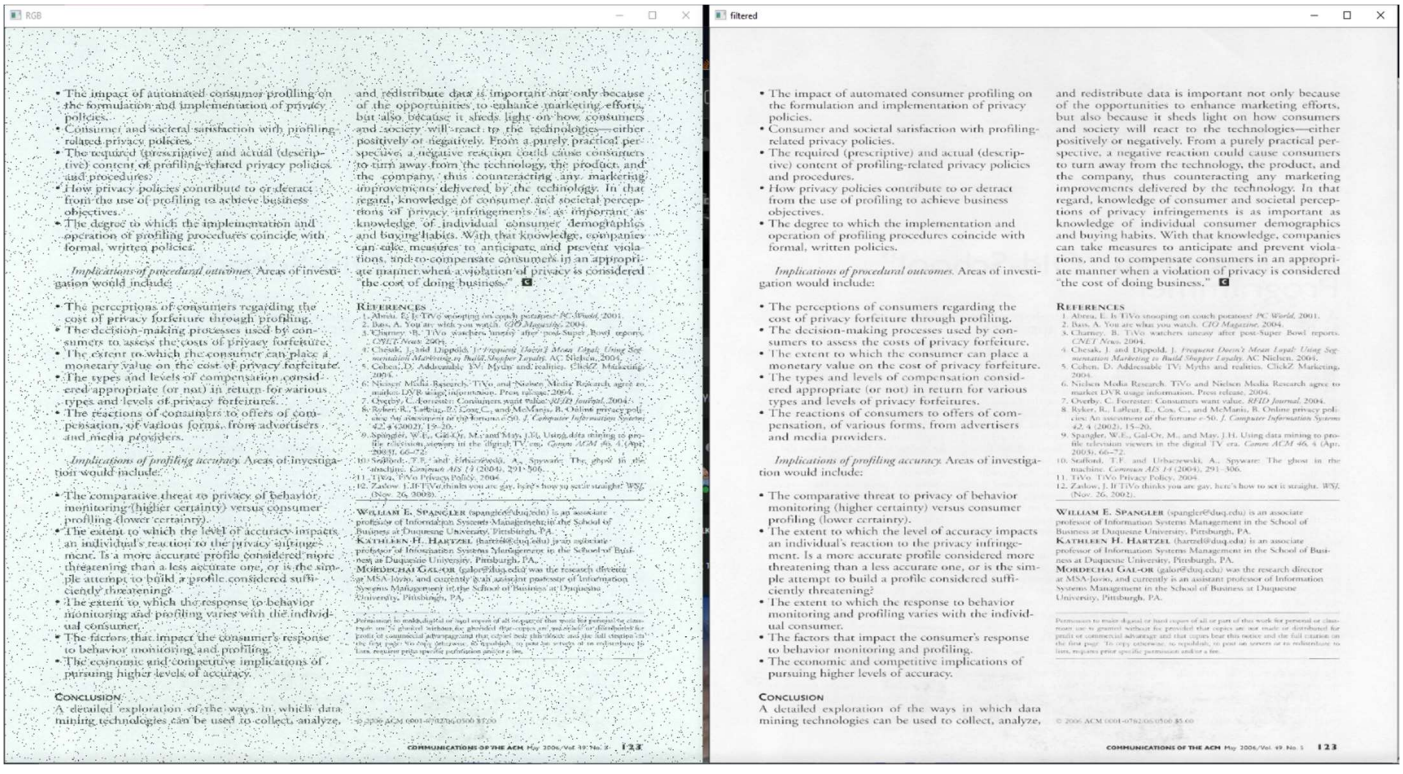


Image 10. Before filtering(left) and after(right) for image 5_noise.png

In conclusion, the filtering results are sufficient enough, even though increased time is required.

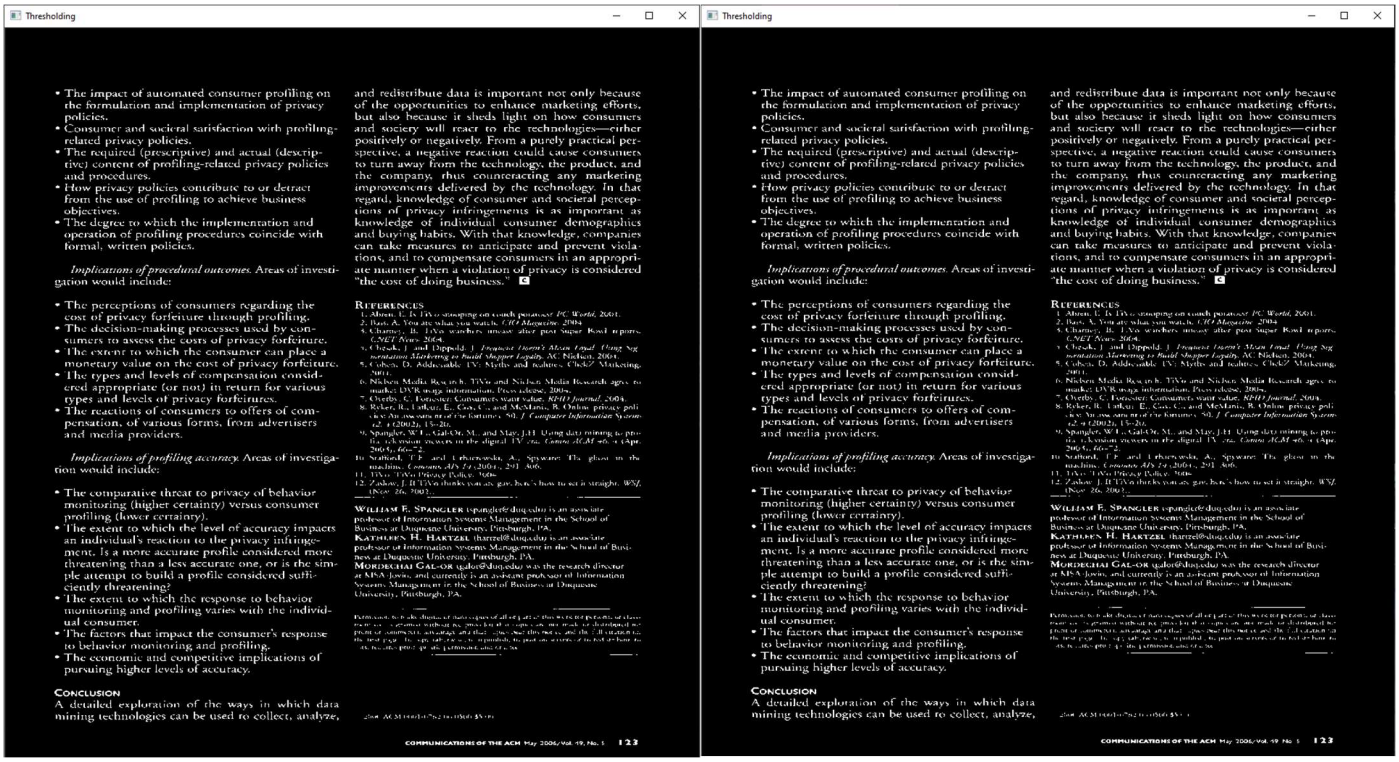
- ***Thresholding***

Before any kind of morphological operation can be done, the image must be transformed into a binary image, in order to differentiate the background with the foreground(images). This is achieved by using the OpenCV library, and with the cv2.threshold command. To avoid the definition of an arbitrary threshold value, which could not lead to optimal results, I used the Otsu threshold, which calculates automatically the value for it, in combination with binary inverted. The Otsu algorithm starts with a beginning threshold of 0 and then tries to find the optimal threshold value by checking the possible values. Then, by using binary inverted, the pixels with value higher than the threshold are assigned equal to zero, while the pixels with lower values than the threshold are assigned equal to one. The results of this method can be seen in the images below.

Images 11,12. Use of threshold technique on images 2_original and 2_noise.png

Images 13,14. Use of threshold technique on images 3_original and 3_noise.png

Images 15,16. Use of threshold technique on images 4_original and 4_noise.png



Images 15,16. Use of threshold technique on images 5_original and 5_noise.png

We can observe that the results are satisfying, since we can differentiate the letters to great extent, no matter the original grayscale values. This is attributed to the use of Otsu's algorithm.

- ***Morphological Operations***

Firsly,I apply dilation, with a relatively large kernel, in order to make each pixel larger, so that neighboring pixels(such as letters, words, sentences) connect with each other and create large, white-colored areas. Due to the binary format of the images, those areas will be easily recognizable. Afterwards, the operation of opening is applied to the previous result, with a kernel of the same size as before, to remove each pixel that was enhanced but is not an area of interest. This way, opening doesn't affect the areas of interest and it operates on possible noise that was enhanced by the previous operation. This conclusion can be easily extracted by taking a look at the results after applying both of those morphological operations.

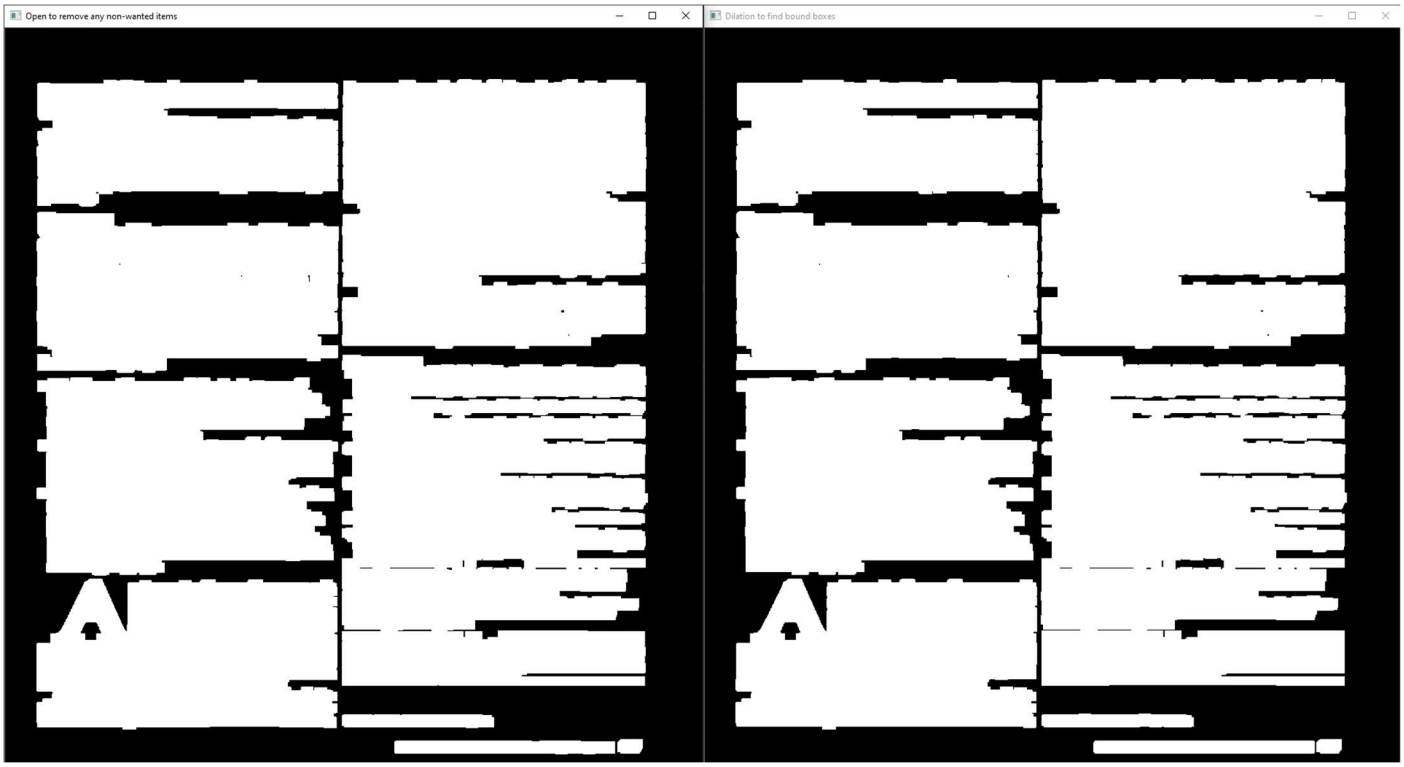


Image 19. Application of dilation(right) and opening(left) on image 2_original.png

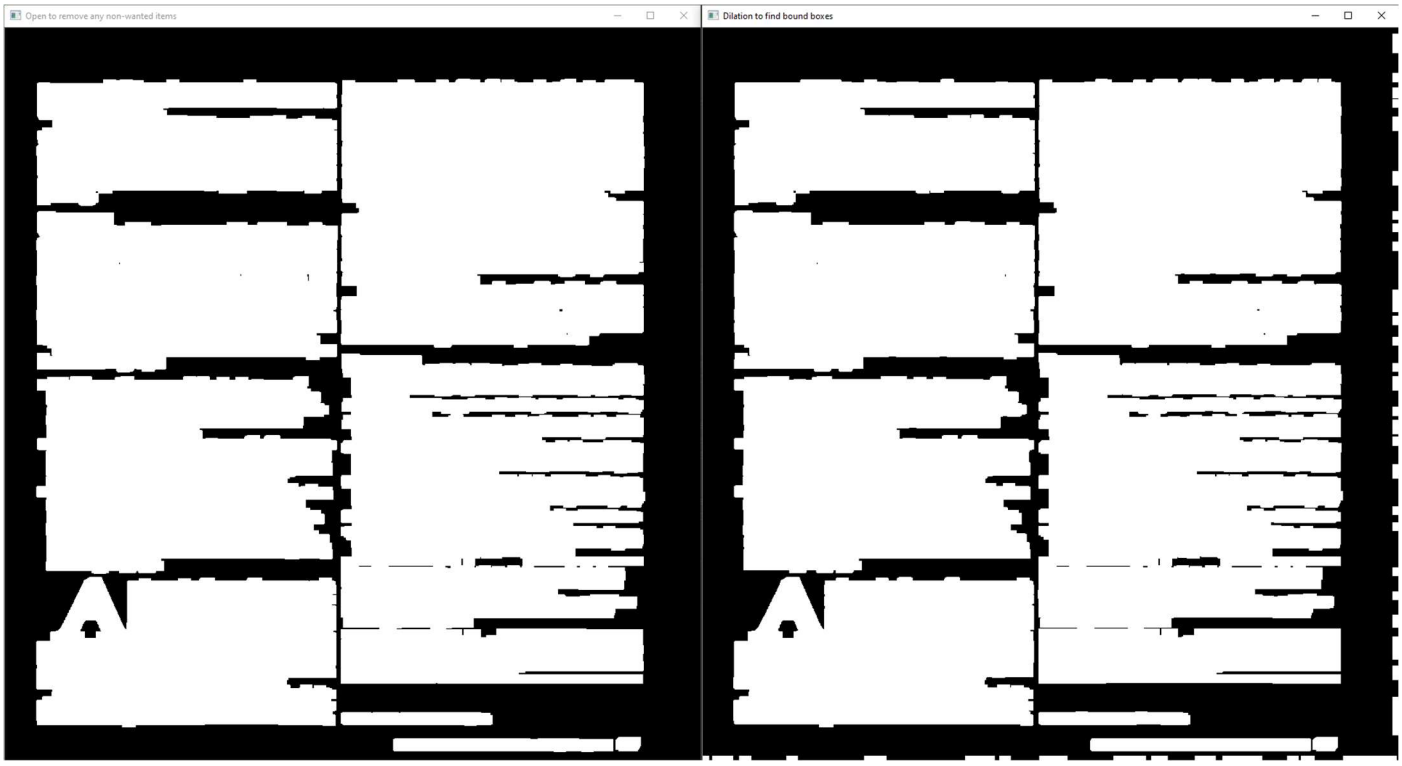


Image 20. Application of dilation(right) and opening(left) on image 2_noise.png

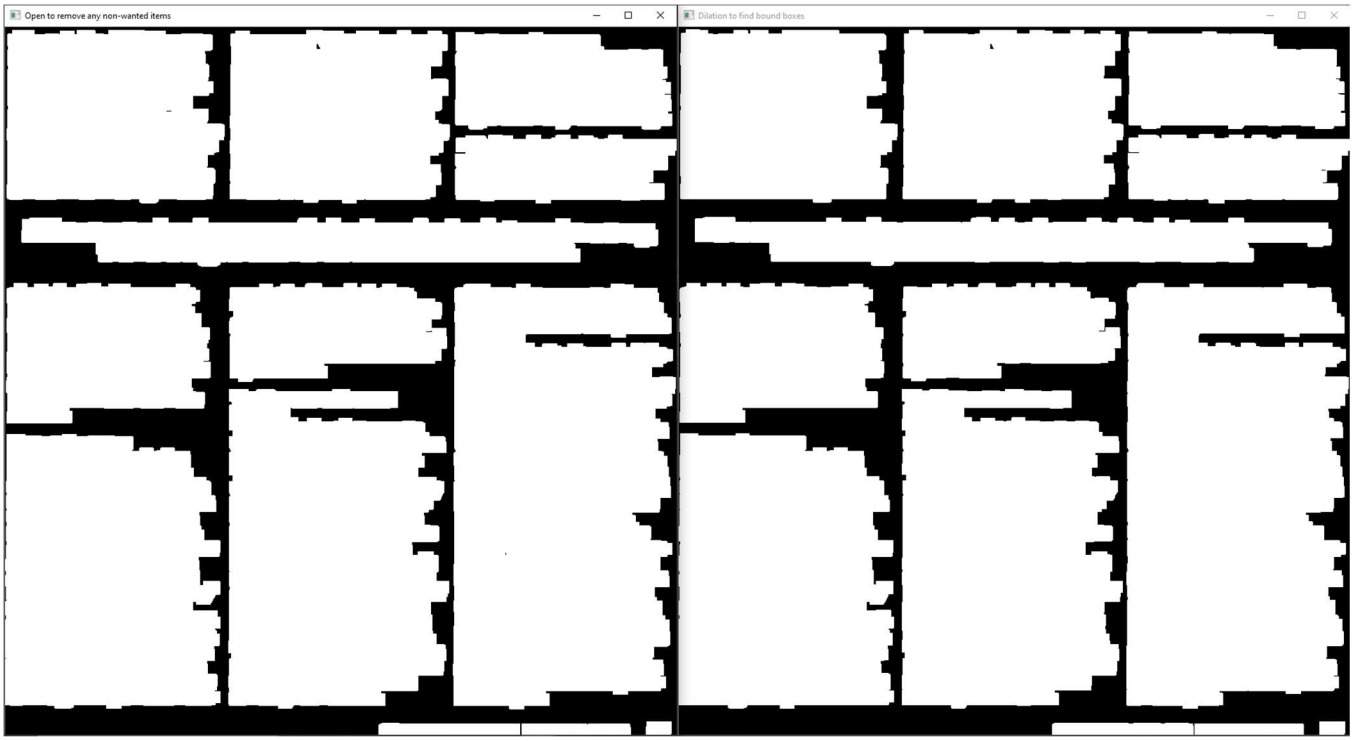


Image 21. Application of dilation(right) and opening(left) on image 3_original.png

Image 22. Application of dilation(right) and opening(left) on image 3_noise.png

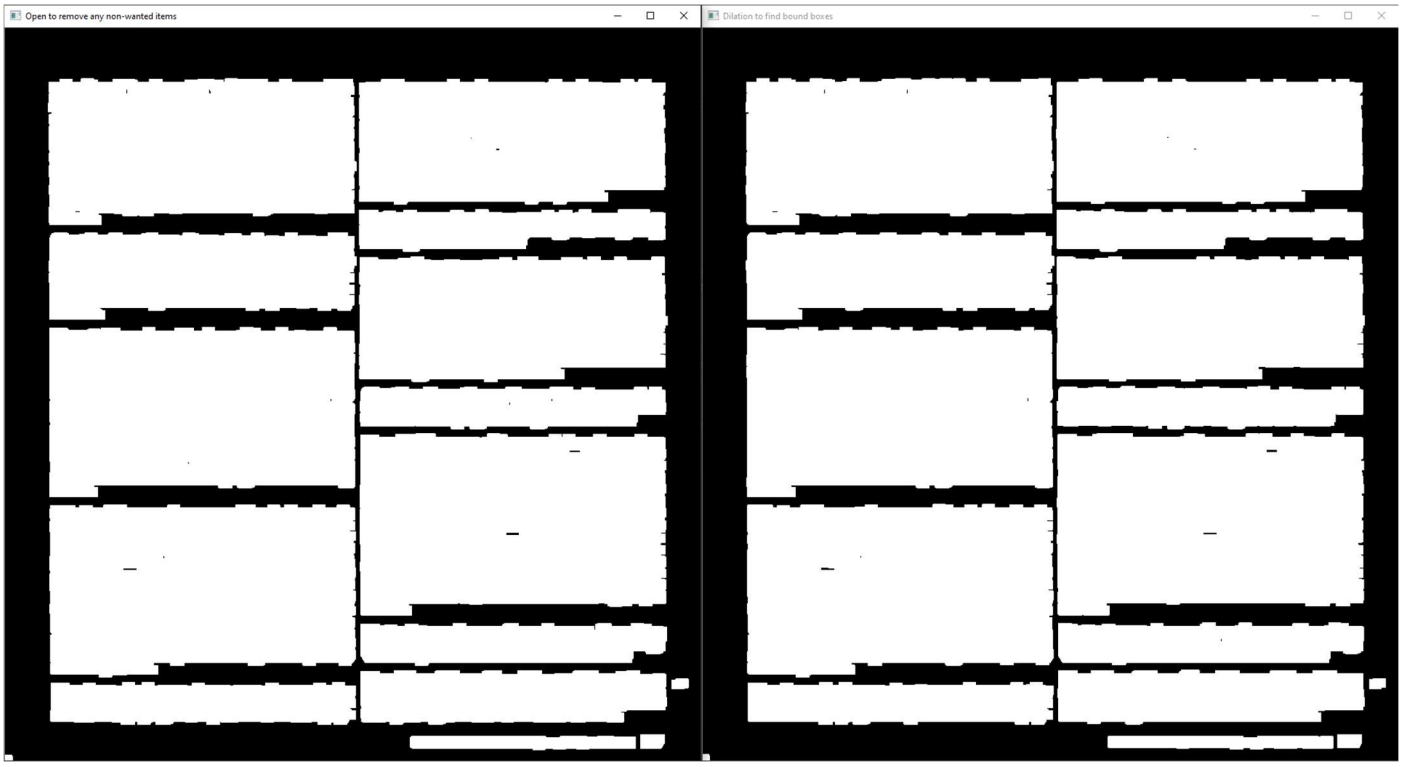


Image 23. Application of dilation(right) and opening(left) on image 4_original.png

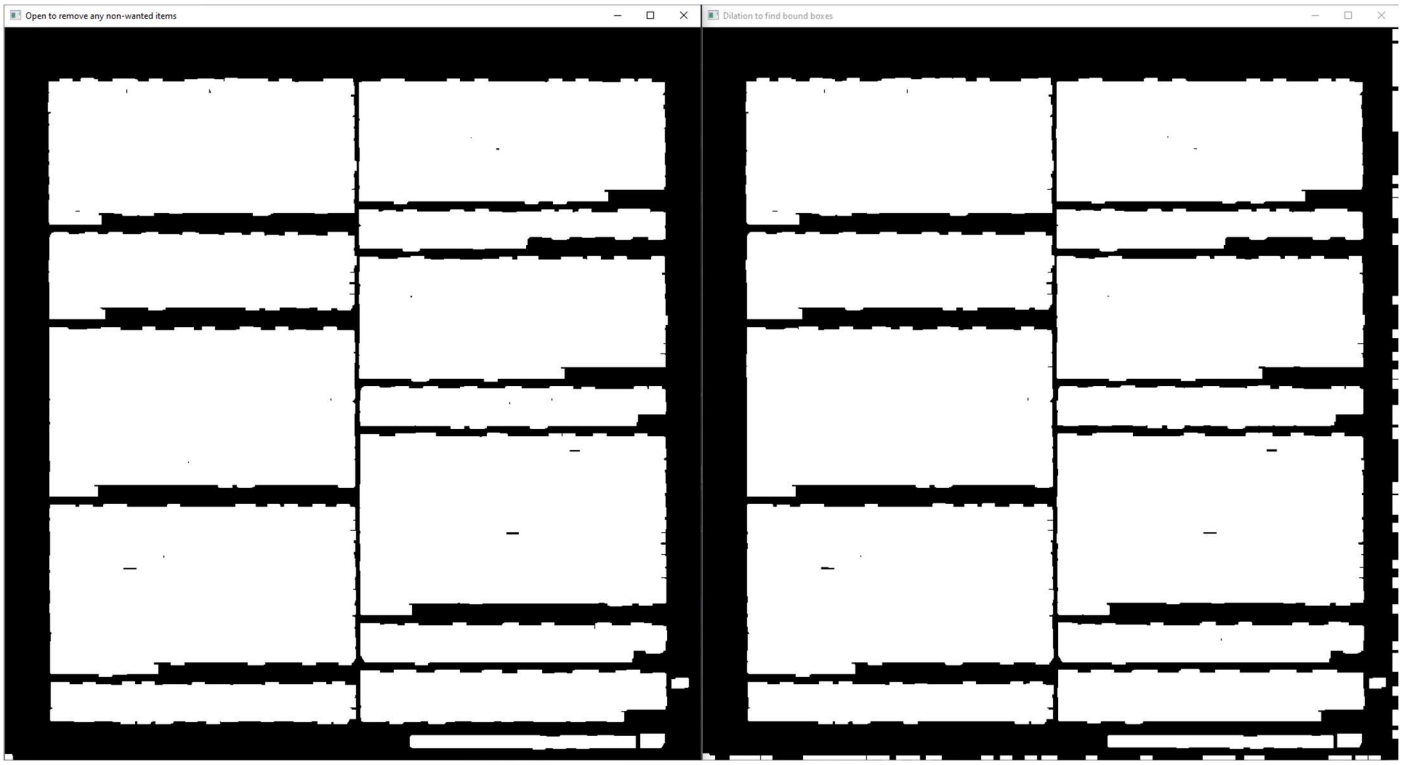


Image 24. Application of dilation(right) and opening(left) on image 4_noise.png

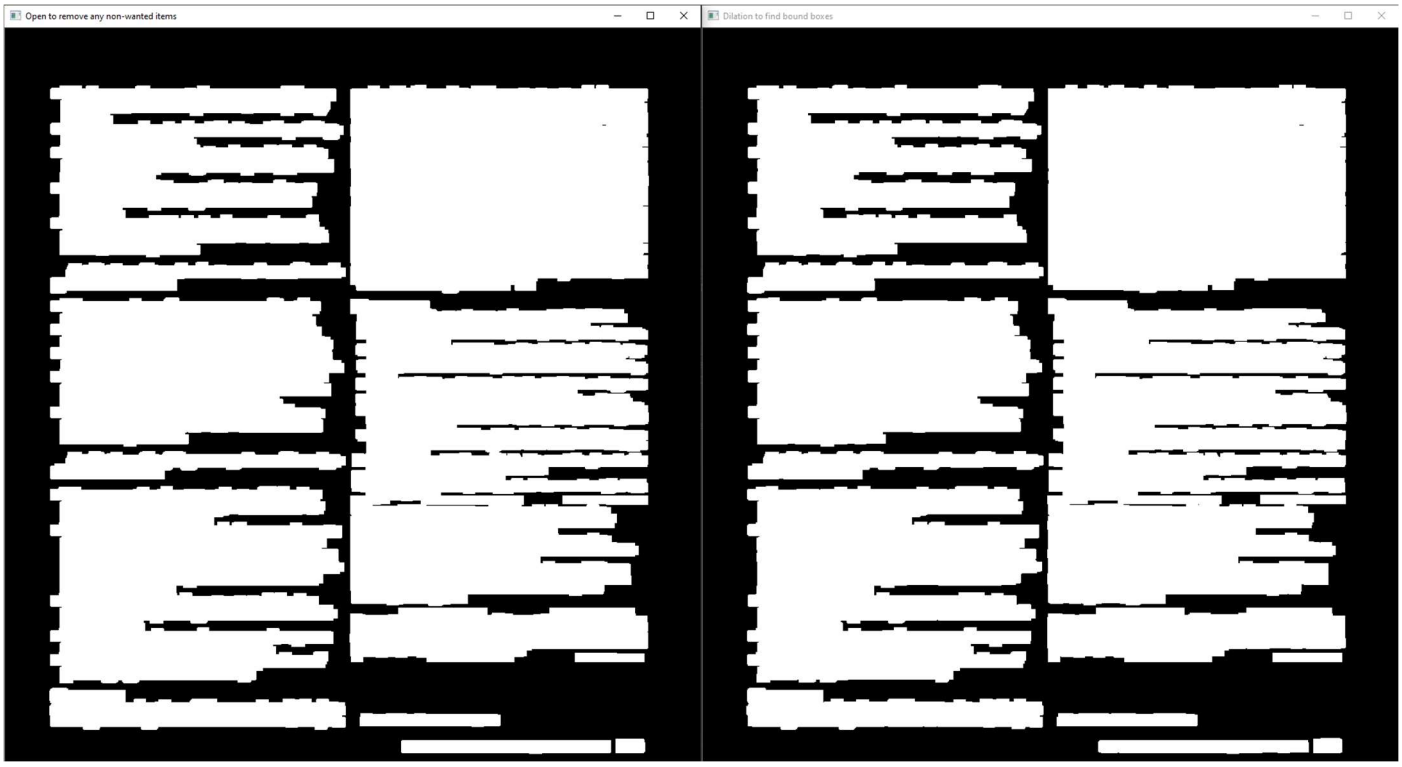


Image 25. Application of dilation(right) and opening(left) on image 5_original.png

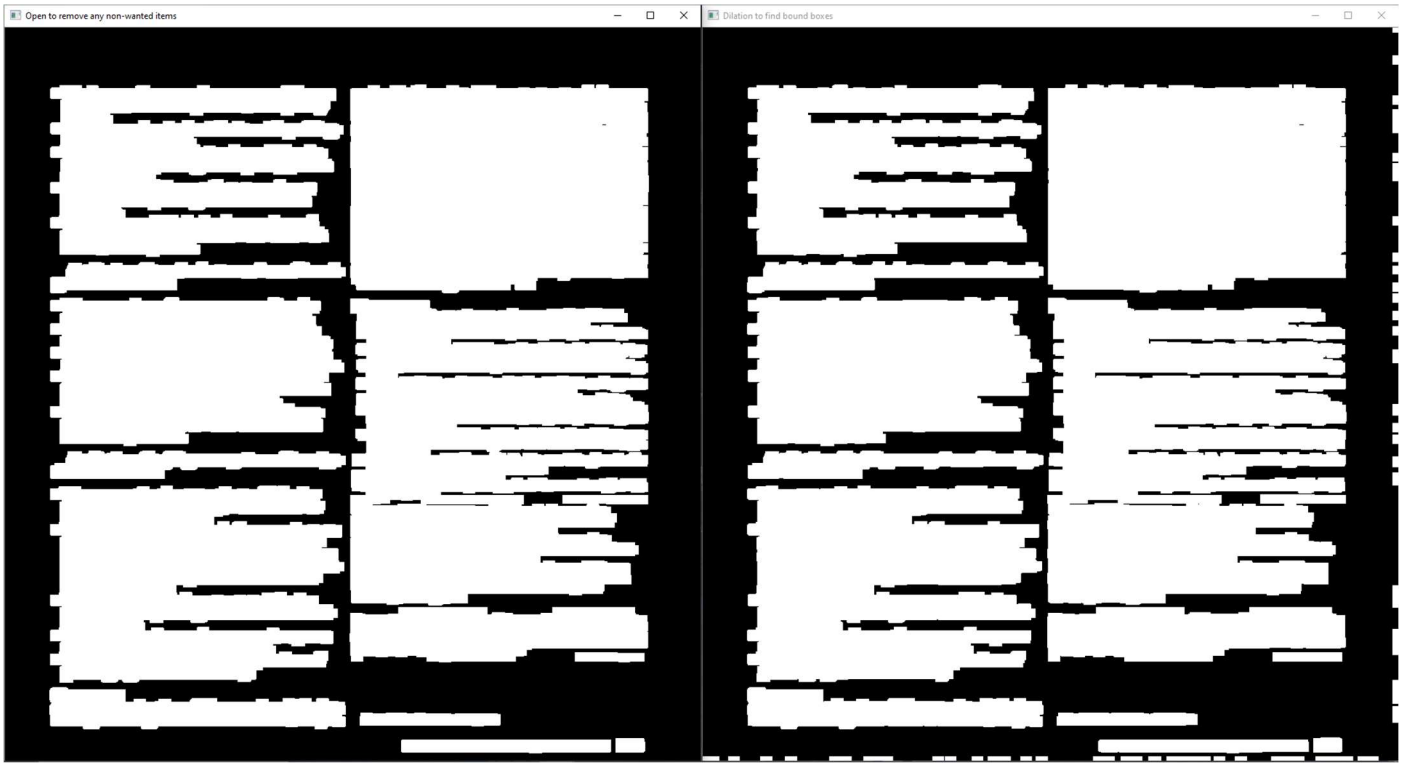


Image 26. Application of dilation(right) and opening(left) on image 5_noise.png

As it can easily be seen, the noisy images had some white spots due to dilation, which were then removed by the use of opening to great extend.

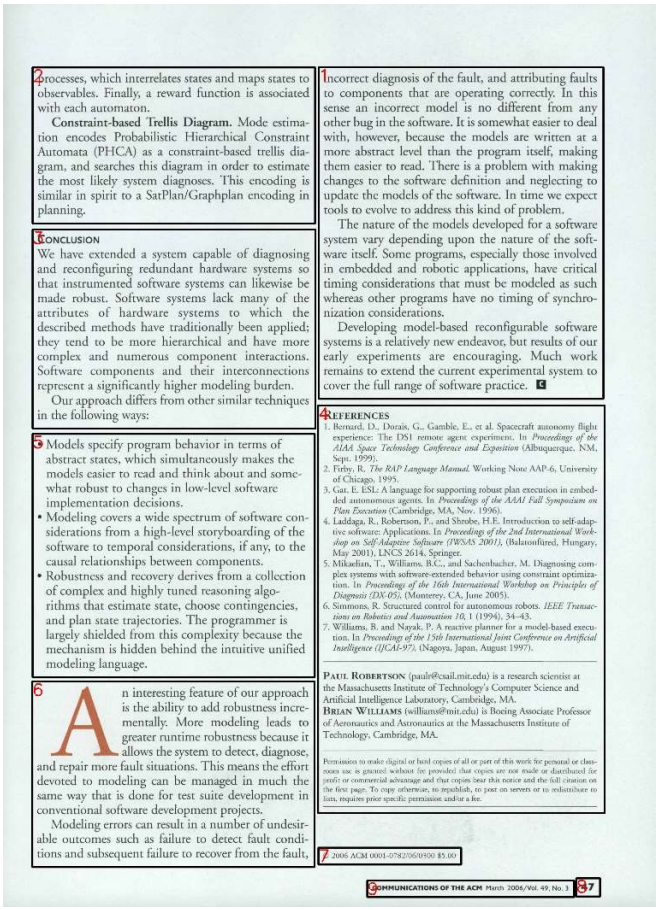
- ***Finding the bounds of each area***

During my preparation for this project, I found two methods, with the one I end up using to be described. I used the cv2.connectedComponentsWithStats, which applies the connected components algorithm to its input.The algorithm, runs its input row by row and in area that pixels are connected with one another, it assigns them a label(a unique number). Then, this process is repeated, now checking if neighboring pixels have different label. If so, they are assigned the lowest valued label out of the possible neighboring labels. Those areas of interested are results of the application the aforementioned morphological operations. Statistics are also returned by this command, such as the surface (measured in pixels) of each area of interest, its start in coordinates of x,y, its height, and its width. So, by using a for loop for every set of statistics created, a bounding box can be created in each area, with a corresponding number at its start.

```
num, _, stats, _ = cv2.connectedComponentsWithStats(morph1)
x = stats[1:, cv2.CC_STAT_LEFT]# 1: to exclude the background,which is labeled 0
y = stats[1:, cv2.CC_STAT_TOP]
w = stats[1:, cv2.CC_STAT_WIDTH]
h = stats[1:, cv2.CC_STAT_HEIGHT]
bdn_area = []
bdn_boxes = []
area = []
endpoint_x = []
endpoint_y = []
words = []
gray_box = []
total_gray = []
mean_gray = []
sums = cv2.integral(gray)#summed area
for i in range(len(x)):
    cv2.rectangle(img, (x[i], y[i]), (x[i]+w[i], y[i]+h[i]), (0,0,0), 5)
    cv2.putText(img, '' + str(i+1), (x[i], y[i] + 50), cv2.FONT_HERSHEY_SIMPLEX, 2, (0, 0, 255), 3)
```

Εικ.27 Υλοποίηση του 1ου ερωτήματος της άσκησης(οι λίστες χρησιμοποιούνται για τα υπόλοιπα ζητούμενα).

Image 27. Implementation of the first part of this project



processes, which interrelates states and maps states to observables. Finally, a reward function is associated with each automaton.

Constraint-based Trellis Diagram. Mode estimation encodes Probabilistic Hierarchical Constraint Automata (PHCA) as a constraint-based trellis diagram, and searches this diagram in order to estimate the most likely system diagnoses. This encoding is similar in spirit to a SatPlan/Graphplan encoding in planning.

Conclusion

We have extended a system capable of diagnosing and reconfiguring redundant hardware systems so that instrumented software systems can likewise be made robust. Software systems lack many of the attributes of hardware systems to which the described methods have traditionally been applied; they tend to be more hierarchical and have more complex and numerous component interactions. Software components and their interconnections represent a significantly higher modeling burden.

Our approach differs from other similar techniques in the following ways:

- Models specify program behavior in terms of abstract states, which simultaneously makes the models easier to read and think about and somewhat robust to changes in low-level software implementation decisions.
- Modeling covers a wide spectrum of software considerations from a high-level storyboarding of the software to temporal considerations, if any, to the causal relationships between components.
- Robustness and recovery derives from a collection of complex and highly tuned reasoning algorithms that estimate state, choose contingencies, and plan state trajectories. The programmer is largely shielded from this complexity because the mechanism is hidden behind the intuitive unified modeling language.

An interesting feature of our approach is the ability to add robustness incrementally. More modeling leads to greater runtime robustness because it allows the system to detect, diagnose, and repair more fault situations. This means the effort devoted to modeling can be managed in much the same way that is done for test suite development in conventional software development projects.

Modeling errors can result in a number of undesirable outcomes such as failure to detect fault conditions and subsequent failure to recover from the fault, incorrect diagnosis of the fault, and attributing faults to components that are operating correctly. In this sense an incorrect model is no different from any other bug in the software. It is somewhat easier to deal with, however, because the models are written at a more abstract level than the program itself, making them easier to read. There is a problem with making changes to the software definition and neglecting to update the models of the software. In time we expect tools to evolve to address this kind of problem.

The nature of the models developed for a software system vary depending upon the nature of the software itself. Some programs, especially those involved in embedded and robotic applications, have critical timing considerations that must be modeled as such whereas other programs have no timing of synchronization considerations.

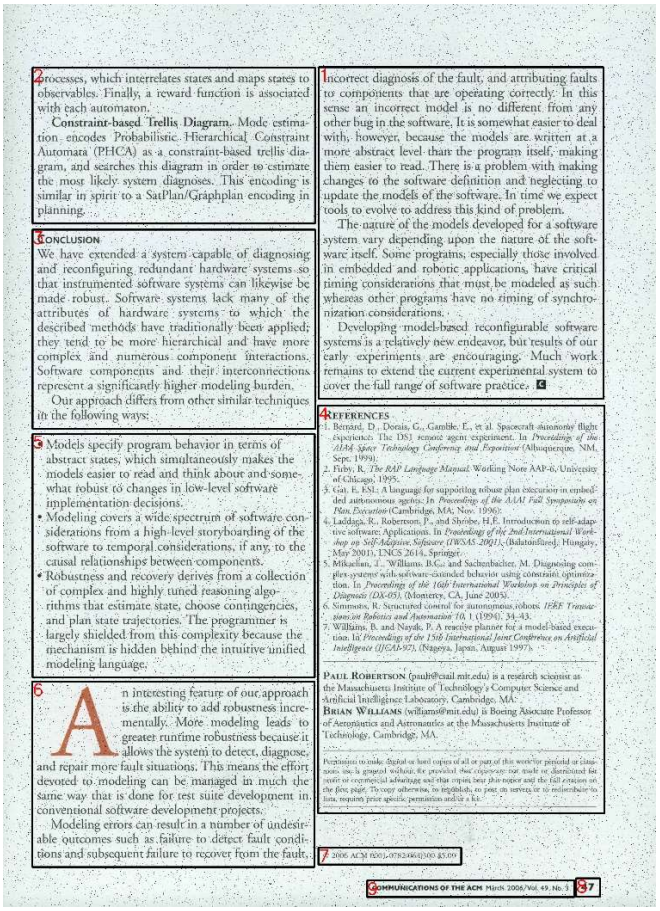
Developing model-based reconfigurable software systems is a relatively new endeavor, but results of our early experiments are encouraging. Much work remains to extend the current experimental system to cover the full range of software practice. C

References

1. Bernard, D., Dorais, G., Gamble, E., et al. Spacecraft autonomy flight experience: The DS1 remote agent experiment. In *Proceedings of the AIAA Space Technology Conference and Exposition* (Albuquerque, NM, Sept. 1999).
2. Firby, R. *The RAP Language Manual.* Working Note AAP-6, University of Chicago, 1995.
3. Gat, E. ESL: A language for supporting robust plan execution in embedded autonomous agents. In *Proceedings of the AAAI Fall Symposium on Plan Execution* (Cambridge, MA, Nov. 1996).
4. Laddaga, R., Robertson, P., and Shrobe, H.E. Introduction to self-adaptive software: Applications. In *Proceedings of the 2nd International Workshop on Self-Adaptive Software (IWSAS 2001)*, (Balatonfüred, Hungary, May 2001), LNCS 2614, Springer.
5. Mikaelian, T., Williams, B.C., and Sachenbacher, M. Diagnosing complex systems with software-extended behavior using constraint optimization. In *Proceedings of the 16th International Workshop on Principles of Diagnosis (DX-05)*, (Monterey, CA, June 2005).
6. Simmons, R. Structured control for autonomous robots. *IEEE Transactions on Robotics and Automation 10*, 1 (1994), 34–43.
7. Williams, B. and Nayak, P. A reactive planner for a model-based executive. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97)*, (Nagoya, Japan, August 1997).

Paul Robertson (paulr@csail.mit.edu) is a research scientist at the Massachusetts Institute of Technology's Computer Science and Artificial Intelligence Laboratory, Cambridge, MA.
Brian Williams (williams@mit.edu) is Boeing Associate Professor of Aeronautics and Astronautics at the Massachusetts Institute of Technology, Cambridge, MA.

© 2006 ACM 0001-0782/06/0900 $5.00



processes, which interrelates states and maps states to observables. Finally, a reward function is associated with each automaton.

Constraint-based Trellis Diagram. Mode estimation encodes Probabilistic Hierarchical Constraint Automata (PHCA) as a constraint-based trellis diagram, and searches this diagram in order to estimate the most likely system diagnoses. This encoding is similar in spirit to a SatPlan/Graphplan encoding in planning.

Conclusion

We have extended a system capable of diagnosing and reconfiguring redundant hardware systems so that instrumented software systems can likewise be made robust. Software systems lack many of the attributes of hardware systems to which the described methods have traditionally been applied; they tend to be more hierarchical and have more complex and numerous component interactions. Software components and their interconnections represent a significantly higher modeling burden.

Our approach differs from other similar techniques in the following ways:

- Models specify program behavior in terms of abstract states, which simultaneously makes the models easier to read and think about and somewhat robust to changes in low-level software implementation decisions.
- Modeling covers a wide spectrum of software considerations from a high-level storyboarding of the software to temporal considerations, if any, to the causal relationships between components.
- Robustness and recovery derives from a collection of complex and highly tuned reasoning algorithms that estimate state, choose contingencies, and plan state trajectories. The programmer is largely shielded from this complexity because the mechanism is hidden behind the intuitive unified modeling language.

An interesting feature of our approach is the ability to add robustness incrementally. More modeling leads to greater runtime robustness because it allows the system to detect, diagnose, and repair more fault situations. This means the effort devoted to modeling can be managed in much the same way that is done for test suite development in conventional software development projects.

Modeling errors can result in a number of undesirable outcomes such as failure to detect fault conditions and subsequent failure to recover from the fault, incorrect diagnosis of the fault, and attributing faults to components that are operating correctly. In this sense an incorrect model is no different from any other bug in the software. It is somewhat easier to deal with, however, because the models are written at a more abstract level than the program itself, making them easier to read. There is a problem with making changes to the software definition and neglecting to update the models of the software. In time we expect tools to evolve to address this kind of problem.

The nature of the models developed for a software system vary depending upon the nature of the software itself. Some programs, especially those involved in embedded and robotic applications, have critical timing considerations that must be modeled as such whereas other programs have no timing of synchronization considerations.

Developing model-based reconfigurable software systems is a relatively new endeavor, but results of our early experiments are encouraging. Much work remains to extend the current experimental system to cover the full range of software practice. C

References

1. Bernard, D., Dorais, G., Gamble, E., et al. Spacecraft autonomy flight experience: The DS1 remote agent experiment. In *Proceedings of the AIAA Space Technology Conference and Exposition* (Albuquerque, NM, Sept. 1999).
2. Firby, R. *The RAP Language Manual.* Working Note AAP-6, University of Chicago, 1995.
3. Gat, E. ESL: A language for supporting robust plan execution in embedded autonomous agents. In *Proceedings of the AAAI Fall Symposium on Plan Execution* (Cambridge, MA, Nov. 1996).
4. Laddaga, R., Robertson, P., and Shrobe, H.E. Introduction to self-adaptive software: Applications. In *Proceedings of the 2nd International Workshop on Self-Adaptive Software (IWSAS 2001)*, (Balatonfüred, Hungary, May 2001), LNCS 2614, Springer.
5. Mikaelian, T., Williams, B.C., and Sachenbacher, M. Diagnosing complex systems with software-extended behavior using constraint optimization. In *Proceedings of the 16th International Workshop on Principles of Diagnosis (DX-05)*, (Monterey, CA, June 2005).
6. Simmons, R. Structured control for autonomous robots. *IEEE Transactions on Robotics and Automation 10*, 1 (1994), 34–43.
7. Williams, B. and Nayak, P. A reactive planner for a model-based executive. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97)*, (Nagoya, Japan, August 1997).

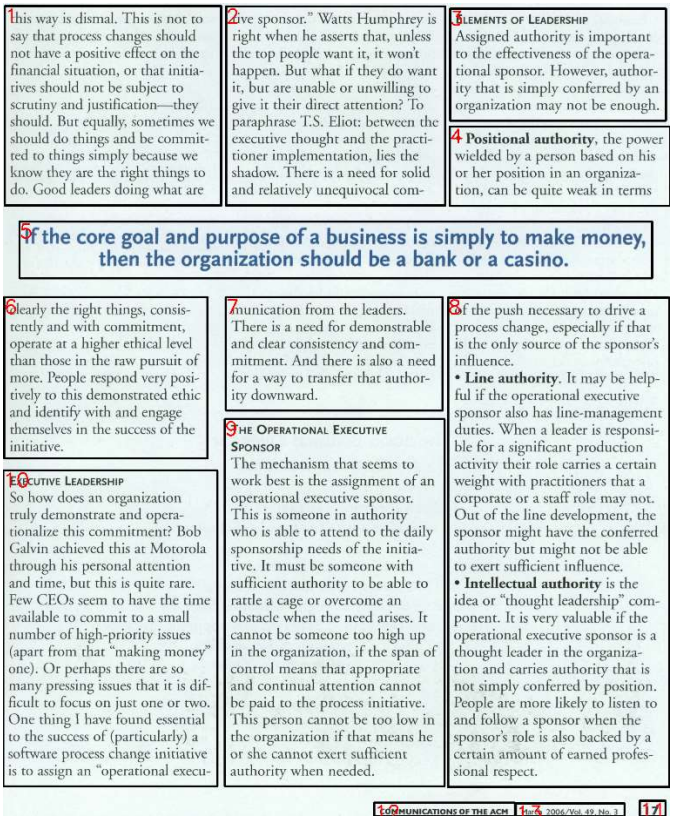
Paul Robertson (paulr@csail.mit.edu) is a research scientist at the Massachusetts Institute of Technology's Computer Science and Artificial Intelligence Laboratory, Cambridge, MA.
Brian Williams (williams@mit.edu) is Boeing Associate Professor of Aeronautics and Astronautics at the Massachusetts Institute of Technology, Cambridge, MA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

© 2006 ACM 0001-0782/06/0900 $5.00

COMMUNICATIONS OF THE ACM March 2006/Vol. 49, No. 3 37

Images 28,29. Bounding boxes on images 2_original.png & 2_noise.png



this way is dismal. This is not to say that process changes should not have a positive effect on the financial situation, or that initiatives should not be subject to scrutiny and justification—they should. But equally, sometimes we should do things and be committed to things simply because we know they are the right things to do. Good leaders doing what are

ive sponsor." Watts Humphrey is right when he asserts that, unless the top people want it, it won't happen. But what if they do want it, but are unable or unwilling to give it their direct attention? To paraphrase T.S. Eliot: between the executive thought and the practitioner implementation, lies the shadow. There is a need for solid and relatively unequivocal com-

Elements of Leadership

Assigned authority is important to the effectiveness of the operational sponsor. However, authority that is simply conferred by an organization may not be enough.

4 Positional authority, the power wielded by a person based on his or her position in an organization, can be quite weak in terms

If the core goal and purpose of a business is simply to make money, then the organization should be a bank or a casino.

learly the right things, consistently and with commitment, operate at a higher ethical level than those in the raw pursuit of more. People respond very positively to this demonstrated ethic and identify with and engage themselves in the success of the initiative.

ecutive Leadership

So how does an organization truly demonstrate and operationalize this commitment? Bob Galvin achieved this at Motorola through his personal attention and time, but this is quite rare. Few CEOs seem to have the time available to commit to a small number of high-priority issues (apart from that "making money" one). Or perhaps there are so many pressing issues that it is difficult to focus on just one or two. One thing I have found essential to the success of (particularly) a software process change initiative is to assign an "operational execu-

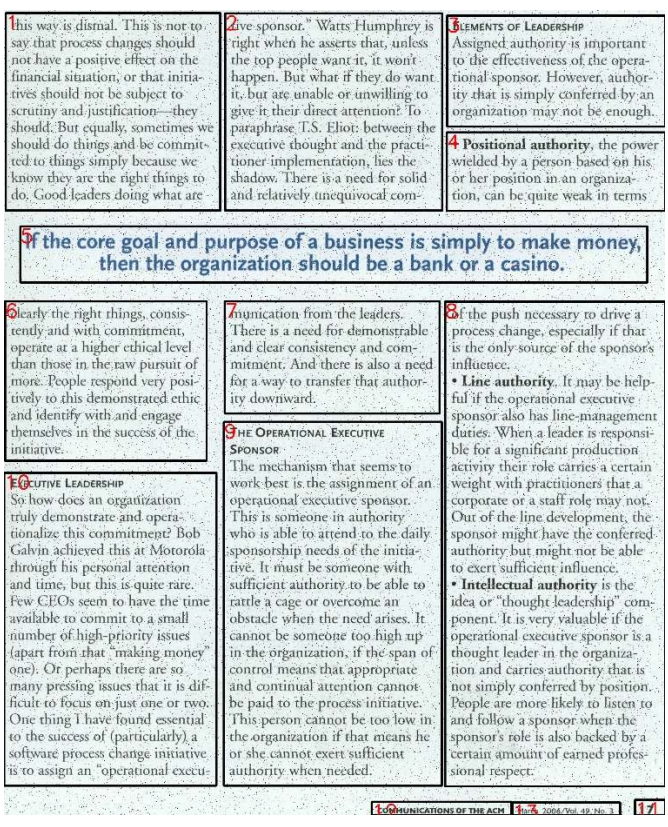
unication from the leaders. There is a need for demonstrable and clear consistency and commitment. And there is also a need for a way to transfer that authority downward.

he Operational Executive Sponsor

The mechanism that seems to work best is the assignment of an operational executive sponsor. This is someone in authority who is able to attend to the daily sponsorship needs of the initiative. It must be someone with sufficient authority to be able to rattle a cage or overcome an obstacle when the need arises. It cannot be someone too high up in the organization, if the span of control means that appropriate and continual attention cannot be paid to the process initiative. This person cannot be too low in the organization if that means he or she cannot exert sufficient authority when needed.

f the push necessary to drive a process change, especially if that is the only source of the sponsor's influence.

- **Line authority**. It may be helpful if the operational executive sponsor also has line-management duties. When a leader is responsible for a significant production activity their role carries a certain weight with practitioners that a corporate or a staff role may not. Out of the line development, the sponsor might have the conferred authority but might not be able to exert sufficient influence.
- **Intellectual authority** is the idea or "thought leadership" component. It is very valuable if the operational executive sponsor is a thought leader in the organization and carries authority that is not simply conferred by position. People are more likely to listen to and follow a sponsor when the sponsor's role is also backed by a certain amount of earned professional respect.



this way is dismal. This is not to say that process changes should not have a positive effect on the financial situation, or that initiatives should not be subject to scrutiny and justification—they should. But equally, sometimes we should do things and be committed to things simply because we know they are the right things to do. Good leaders doing what are

ive sponsor." Watts Humphrey is right when he asserts that, unless the top people want it, it won't happen. But what if they do want it, but are unable or unwilling to give it their direct attention? To paraphrase T.S. Eliot: between the executive thought and the practitioner implementation, lies the shadow. There is a need for solid and relatively unequivocal com-

Elements of Leadership

Assigned authority is important to the effectiveness of the operational sponsor. However, authority that is simply conferred by an organization may not be enough.

4 Positional authority, the power wielded by a person based on his or her position in an organization, can be quite weak in terms

If the core goal and purpose of a business is simply to make money, then the organization should be a bank or a casino.

learly the right things, consistently and with commitment, operate at a higher ethical level than those in the raw pursuit of more. People respond very positively to this demonstrated ethic and identify with and engage themselves in the success of the initiative.

ecutive Leadership

So how does an organization truly demonstrate and operationalize this commitment? Bob Galvin achieved this at Motorola through his personal attention and time, but this is quite rare. Few CEOs seem to have the time available to commit to a small number of high-priority issues (apart from that "making money" one). Or perhaps there are so many pressing issues that it is difficult to focus on just one or two. One thing I have found essential to the success of (particularly) a software process change initiative is to assign an "operational execu-

unication from the leaders. There is a need for demonstrable and clear consistency and commitment. And there is also a need for a way to transfer that authority downward.

he Operational Executive Sponsor

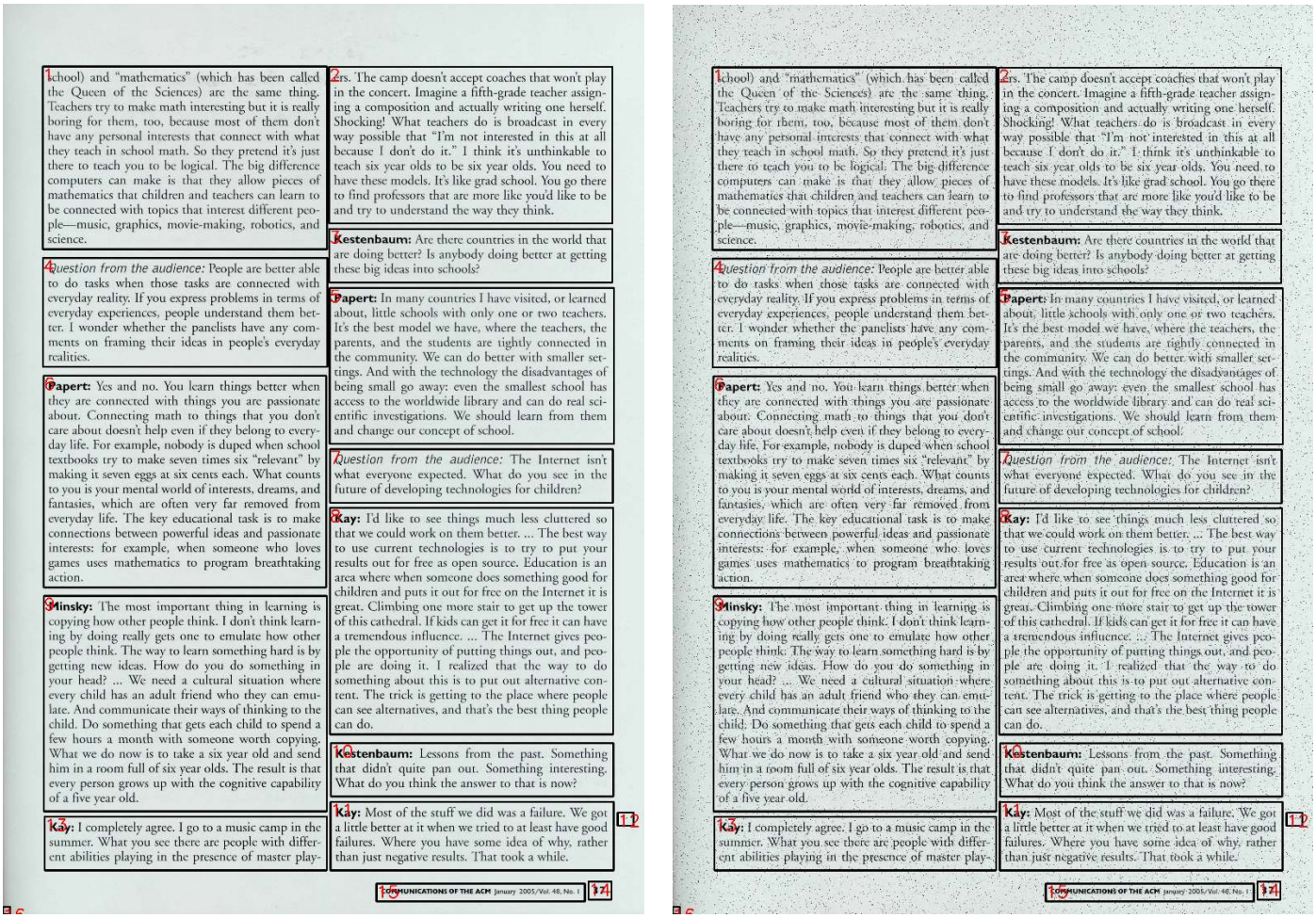
The mechanism that seems to work best is the assignment of an operational executive sponsor. This is someone in authority who is able to attend to the daily sponsorship needs of the initiative. It must be someone with sufficient authority to be able to rattle a cage or overcome an obstacle when the need arises. It cannot be someone too high up in the organization, if the span of control means that appropriate and continual attention cannot be paid to the process initiative. This person cannot be too low in the organization if that means he or she cannot exert sufficient authority when needed.

f the push necessary to drive a process change, especially if that is the only source of the sponsor's influence.

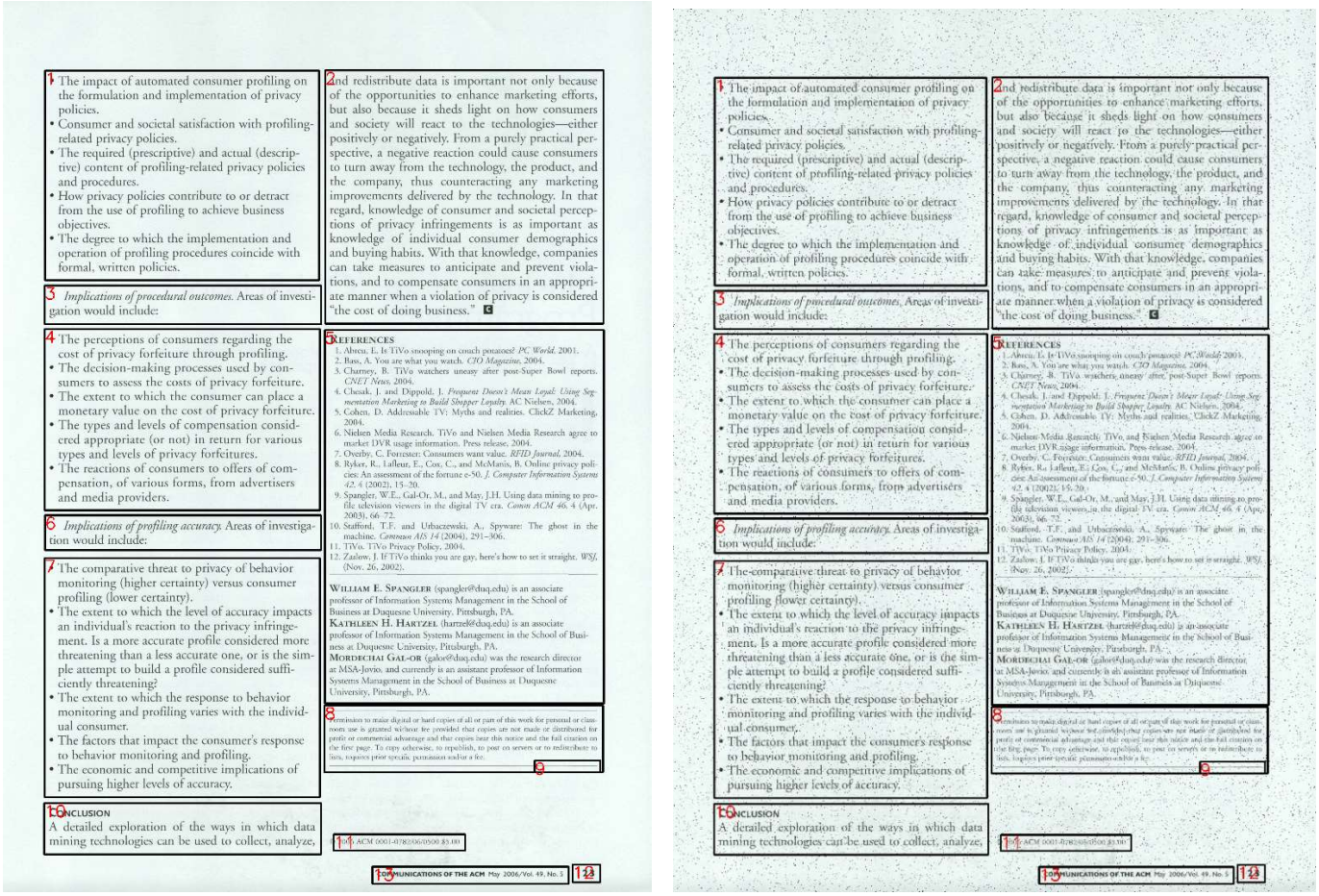
- **Line authority**. It may be helpful if the operational executive sponsor also has line-management duties. When a leader is responsible for a significant production activity their role carries a certain weight with practitioners that a corporate or a staff role may not. Out of the line development, the sponsor might have the conferred authority but might not be able to exert sufficient influence.
- **Intellectual authority** is the idea or "thought leadership" component. It is very valuable if the operational executive sponsor is a thought leader in the organization and carries authority that is not simply conferred by position. People are more likely to listen to and follow a sponsor when the sponsor's role is also backed by a certain amount of earned professional respect.

COMMUNICATIONS OF THE ACM March 2006/Vol. 49, No. 3 17

Images 30,31. Bounding boxes on images 3_original.png & 3_noise.png



Images 32,33. Bounding boxes on images 4_original.png & 4_noise.png



Images 34,35. Bounding boxes on images 5_original.png & 5_noise.png

This implementation is pretty accurate, apart from some wrong areas that are detected on the 4 and 5 images set. Efficiency wise, the implementation has:

1. 100% accuracy for the image sets 2,3
2. For 4_original και 4_noise images 2 more areas are detected (#12,#16). Firstly, a pixel (existing and in the original image)at the lower right part of the image is not properly filtered, so in turn it gets to be considered as an area of interest. Also, at the lower left part, another small pixel is not filtered and still is considered as an area of interest.
3. For the images 5_original and 5_noise, one more area is detected (#9),which is due to a small line shading lower right inside area #8, which is not enlarged enough to be considered part of area #8.

Therefore, even though the implementation finds some areas more, the fact that the extra areas appear on both the original and noisy ones could mean that this is a drawback of the text they represent. Still, I consider the results successful. The aforementioned failures could be avoided by using different morphological operations, different kernel size or different filter and implementation in general.

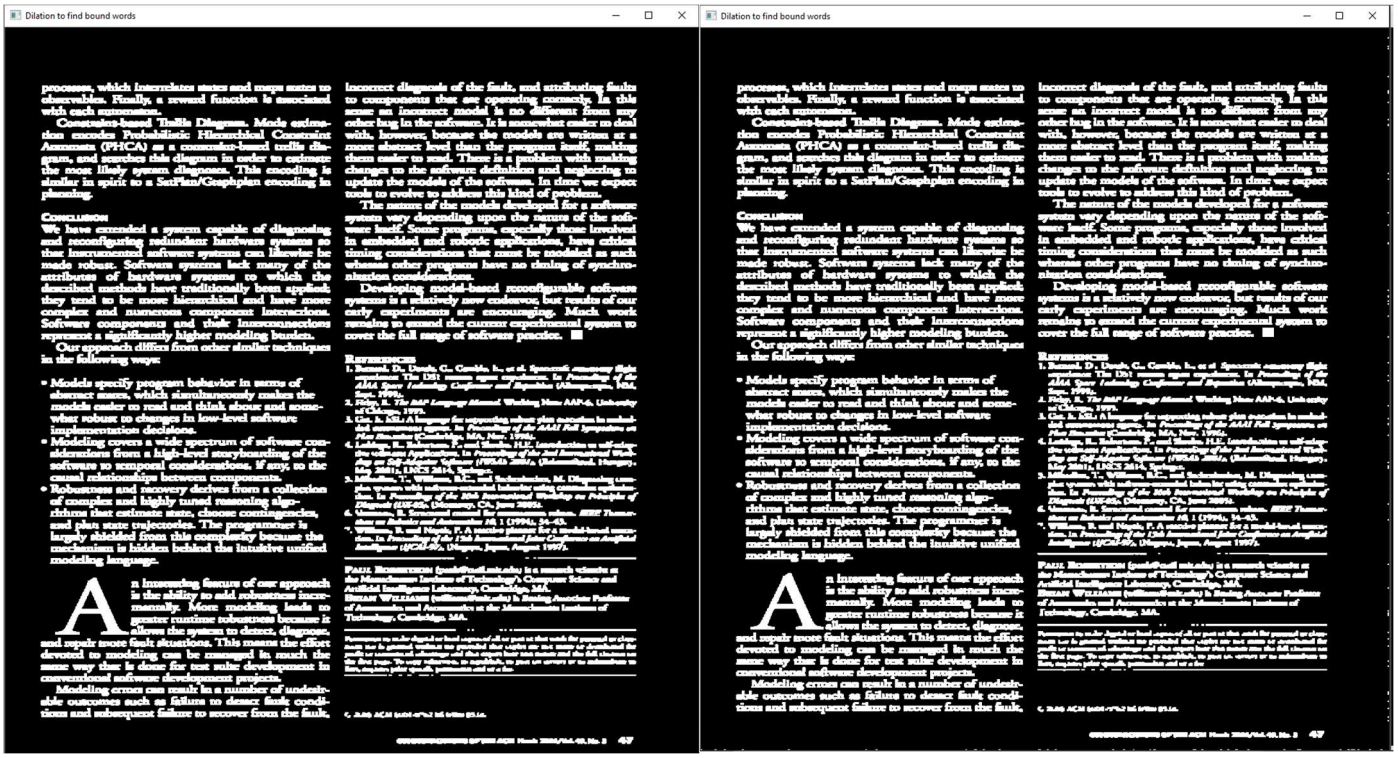
- ***Finding the rest of the desired features***

```
endpoint_x.insert(i, x[i]+w[i])
endpoint_y.insert(i, y[i]+h[i])
bdn_boxes.insert(i, bin2[y[i]:endpoint_y[i], x[i]:endpoint_x[i]])#crop binary img according to boundary boxes
nums, _, _, _ = cv2.connectedComponentsWithStats(morph2[y[i]:endpoint_y[i], x[i]:endpoint_x[i]])#connectedComp to find num of labels,i.e. words
words.insert(i, nums-1)#minus-1 cause background gets a label too
area.insert(i, cv2.countNonZero(bdn_boxes[i]))#finding which pixels have non zero value,due to bin inv
bdn_area.insert(i, w[i] * h[i])#total box area
gray_box.insert(i,gray[y[i]:endpoint_y[i], x[i]:endpoint_x[i]])#crop grayscale img according to boundary boxes,to find mean gray value
total_gray.insert(i,(w[i],h[i]))
mean_gray.insert(i,(sums[total_gray[i]]/(bdn_area[i])))#mean gray for each sub-area
print('Square for area ' + str(i+1), ' is ' + str(bdn_area[i]),' pixels, while text area is '+str(area[i]),
      'pixels,word approximated count is ' +str(words[i]),'and mean gray value is '+str(mean_gray[i]))
```

Images 36. The implementation for the rest of the desired features

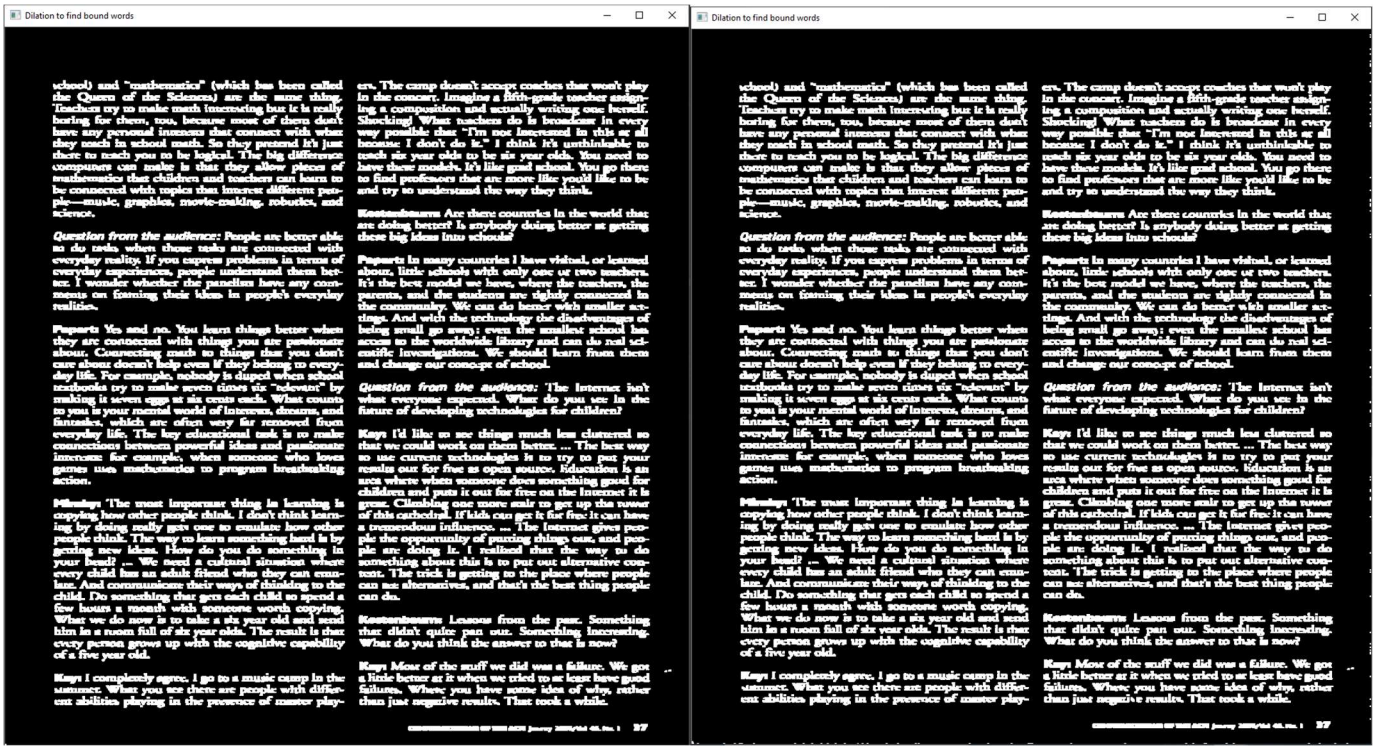
The area of each bounding box is calculated by multiplying its height with its width, which are then saved at the list bdn_area. For the computation of the rest of the desired features, the dimensions of each box are saved in the list bdn_boxes, and then those are applied onto the images that results from thresholding using another dilation to find the number of words in each area. This way is fast, as it needs one small for-loop. To find the area that is used by letters only, one has to simple count the number of pixels with value equal to 255 (or 1 in binary terms) in each area of interest. To find the words, a smaller dilation kernel is needed, sized here 7x7, but big enough to connect the letters of each word together. So, after applying the morphological operation, connectedComponentsWithStats is used on the binary images. This stems from the logic that each connected component is a word. So, the number of areas is equal to the number of words. Applying this method to each area of interest,

the number of words can be found in each bounding box. From the aforementioned description of this method, it can be easily concluded that the number of words is dependent on the kernel size used in the morphological operations. Finally, for the last requested feature, I applied the previous idea in the grayscale image, where the method of the Summed Area Table (or Integral Image) is applied. According to this algorithm, each pixel value is a sum of its own value with the values of the pixels before this one in the horizontal axis and in the vertical axis, subtracting by it the value of the upper left diagonal pixel value. Progressively, the result is that the last element of this array corresponds to the last pixel of each area of interest, so its value is equal to the sum of all the values. Diving then by each area, the mean grayscale value is calculated. This process is repeated for every bounding box area.

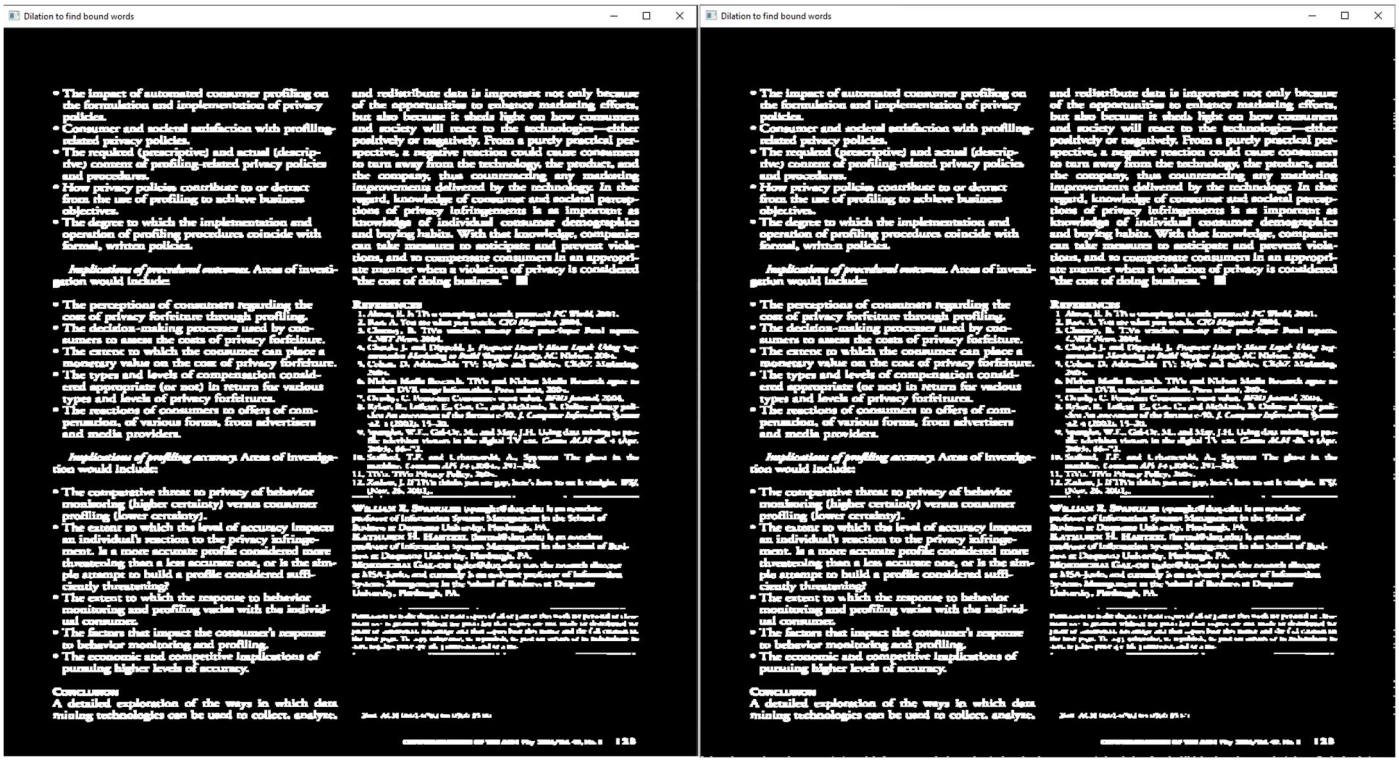


Images 37,38. Dilation to find words in images 2_original.png & 2_noise.png

Images 39,40. Dilation to find words in images 3_original.png & 3_noise.png



Images 41,42. Dilation to find words in images 4_original.png & 4_noise.png



Images 43,44. Dilation to find words in images 5_original.png & 5_noise.png

An obstacle that appeared was when calculating the number of words. Sometimes, dots, for example in the dot in the letter "i", was wrongly counted as a word. The same thing applies for punctuation marks, because those were enlarged during dilation. I concluded that in smaller bounding boxes, due to dilation, some words are missed, while in larger areas, some more words are counted. Also,in case a pixel escapes the filtering and thresholding process, it could be considered as a word, though I did not face such problems.Once again, using different kernel sizes or different morphological operations coud solve some of these problems. Below are the results for each image.

```
Square for area 1  is 1201760  pixels, while text area is 162207 pixels,word approximated count is 207 and mean gray value is 225.24943249900147
Square for area 2  is 560899  pixels, while text area is 66743 pixels,word approximated count is 85 and mean gray value is 225.86683342277308
Square for area 3  is 715680  pixels, while text area is 83412 pixels,word approximated count is 102 and mean gray value is 225.51962050078248
Square for area 4  is 1485988  pixels, while text area is 147306 pixels,word approximated count is 343 and mean gray value is 224.07107325227392
Square for area 5  is 874214  pixels, while text area is 99929 pixels,word approximated count is 143 and mean gray value is 225.10095468615236
Square for area 6  is 665972  pixels, while text area is 88153 pixels,word approximated count is 111 and mean gray value is 225.67443225841328
Square for area 7  is 30540  pixels, while text area is 1816 pixels,word approximated count is 8 and mean gray value is 226.68690242305175
Square for area 8  is 5525  pixels, while text area is 628 pixels,word approximated count is 1 and mean gray value is 226.98135746606334
Square for area 9  is 43542  pixels, while text area is 5805 pixels,word approximated count is 9 and mean gray value is 226.69900326121905
```

Image 45. Results for image 2_original.png

```
Square for area 1  is 1201760  pixels, while text area is 161582 pixels,word approximated count is 209 and mean gray value is 225.31916272799893
Square for area 2  is 560899  pixels, while text area is 66475 pixels,word approximated count is 86 and mean gray value is 225.9475003521133
Square for area 3  is 716688  pixels, while text area is 82975 pixels,word approximated count is 103 and mean gray value is 225.60371877302256
Square for area 4  is 1485555  pixels, while text area is 144974 pixels,word approximated count is 353 and mean gray value is 224.12291702427711
Square for area 5  is 874214  pixels, while text area is 99217 pixels,word approximated count is 146 and mean gray value is 225.17739134811384
Square for area 6  is 665972  pixels, while text area is 87676 pixels,word approximated count is 111 and mean gray value is 225.7585153730187
Square for area 7  is 30540  pixels, while text area is 1773 pixels,word approximated count is 8 and mean gray value is 226.68726260641782
Square for area 8  is 5525  pixels, while text area is 628 pixels,word approximated count is 1 and mean gray value is 226.98425339366517
Square for area 9  is 43601  pixels, while text area is 5765 pixels,word approximated count is 9 and mean gray value is 226.70177289511707
```

Image 46. Results for image 2_noise.png

```
Square for area 1  is 402549  pixels, while text area is 46492 pixels,word approximated count is 90 and mean gray value is 222.211042630835
Square for area 2  is 406752  pixels, while text area is 48432 pixels,word approximated count is 92 and mean gray value is 222.4153587443946
Square for area 3  is 225378  pixels, while text area is 26752 pixels,word approximated count is 40 and mean gray value is 221.8444568680173
Square for area 4  is 162866  pixels, while text area is 19220 pixels,word approximated count is 35 and mean gray value is 222.01030908845308
Square for area 5  is 334236  pixels, while text area is 57589 pixels,word approximated count is 36 and mean gray value is 222.01496547349777
Square for area 6  is 305532  pixels, while text area is 34377 pixels,word approximated count is 63 and mean gray value is 220.60050993022008
Square for area 7  is 229052  pixels, while text area is 23129 pixels,word approximated count is 39 and mean gray value is 221.92955748039748
Square for area 8  is 1009125  pixels, while text area is 122867 pixels,word approximated count is 200 and mean gray value is 224.0286287625418
Square for area 9  is 752862  pixels, while text area is 84909 pixels,word approximated count is 150 and mean gray value is 223.5328413440976
Square for area 10  is 631110  pixels, while text area is 74031 pixels,word approximated count is 135 and mean gray value is 223.33343315745273
Square for area 11  is 3713  pixels, while text area is 446 pixels,word approximated count is 2 and mean gray value is 227.11069216267168
Square for area 12  is 17630  pixels, while text area is 3855 pixels,word approximated count is 4 and mean gray value is 227.50124787294385
Square for area 13  is 12948  pixels, while text area is 1653 pixels,word approximated count is 5 and mean gray value is 229.209453197405
```

Image 47. Results for image 3_original.png

```
Square for area 1  is 402549  pixels, while text area is 46210 pixels,word approximated count is 91 and mean gray value is 222.3082730301156
Square for area 2  is 406752  pixels, while text area is 48159 pixels,word approximated count is 93 and mean gray value is 222.5117688419479
Square for area 3  is 229824  pixels, while text area is 26545 pixels,word approximated count is 45 and mean gray value is 222.29887218045113
Square for area 4  is 162866  pixels, while text area is 19102 pixels,word approximated count is 39 and mean gray value is 222.1080704382744
Square for area 5  is 334236  pixels, while text area is 57560 pixels,word approximated count is 36 and mean gray value is 222.12286528081955
Square for area 6  is 305532  pixels, while text area is 34113 pixels,word approximated count is 64 and mean gray value is 220.70669520704868
Square for area 7  is 229052  pixels, while text area is 23078 pixels,word approximated count is 40 and mean gray value is 222.02596790248504
Square for area 8  is 1013610  pixels, while text area is 121922 pixels,word approximated count is 218 and mean gray value is 224.19433213958033
Square for area 9  is 753984  pixels, while text area is 84280 pixels,word approximated count is 149 and mean gray value is 223.6639822595705
Square for area 10  is 631110  pixels, while text area is 73433 pixels,word approximated count is 136 and mean gray value is 223.426965188319
Square for area 11  is 4371  pixels, while text area is 447 pixels,word approximated count is 4 and mean gray value is 228.57332418210936
Square for area 12  is 17630  pixels, while text area is 3844 pixels,word approximated count is 6 and mean gray value is 227.5770845150312
Square for area 13  is 12948  pixels, while text area is 1636 pixels,word approximated count is 6 and mean gray value is 229.2220420142107
```

Image 48. Results for image 3_noise.png

```
Square for area 1  is 652536  pixels, while text area is 64651 pixels,word approximated count is 138 and mean gray value is 214.4488871112092
Square for area 2  is 556665  pixels, while text area is 57488 pixels,word approximated count is 135 and mean gray value is 214.74106688942183
Square for area 3  is 190380  pixels, while text area is 17126 pixels,word approximated count is 35 and mean gray value is 217.75938123752496
Square for area 4  is 388702  pixels, while text area is 36165 pixels,word approximated count is 74 and mean gray value is 215.45757675545792
Square for area 5  is 556871  pixels, while text area is 56611 pixels,word approximated count is 128 and mean gray value is 214.72619511520622
Square for area 6  is 751751  pixels, while text area is 77617 pixels,word approximated count is 164 and mean gray value is 214.36657882729787
Square for area 7  is 190380  pixels, while text area is 17255 pixels,word approximated count is 38 and mean gray value is 217.75938123752496
Square for area 8  is 805409  pixels, while text area is 81086 pixels,word approximated count is 228 and mean gray value is 214.14589481865735
Square for area 9  is 765528  pixels, while text area is 80802 pixels,word approximated count is 185 and mean gray value is 214.28917035039868
Square for area 10  is 189189  pixels, while text area is 18538 pixels,word approximated count is 35 and mean gray value is 217.83616383616385
Square for area 11  is 243729  pixels, while text area is 23274 pixels,word approximated count is 62 and mean gray value is 216.65930603251974
Square for area 12  is 2632  pixels, while text area is 22 pixels,word approximated count is 2 and mean gray value is 210.2644376899696
Square for area 13  is 189810  pixels, while text area is 18460 pixels,word approximated count is 42 and mean gray value is 217.77300985195723
Square for area 14  is 5184  pixels, while text area is 519 pixels,word approximated count is 2 and mean gray value is 212.1454475308642
Square for area 15  is 46494  pixels, while text area is 5141 pixels,word approximated count is 9 and mean gray value is 213.17348475072052
Square for area 16  is 675  pixels, while text area is 24 pixels,word approximated count is 1 and mean gray value is 207.95703703703703
```

Image 49. Results for image 4_original.png

```
Square for area 1  is 652536  pixels, while text area is 64155 pixels,word approximated count is 139 and mean gray value is 214.50970980911399
Square for area 2  is 556665  pixels, while text area is 56991 pixels,word approximated count is 136 and mean gray value is 214.81177548435772
Square for area 3  is 190380  pixels, while text area is 17042 pixels,word approximated count is 36 and mean gray value is 217.76937703540287
Square for area 4  is 388702  pixels, while text area is 35866 pixels,word approximated count is 74 and mean gray value is 215.52207346501947
Square for area 5  is 557878  pixels, while text area is 56172 pixels,word approximated count is 132 and mean gray value is 214.7889610273214
Square for area 6  is 751751  pixels, while text area is 76984 pixels,word approximated count is 166 and mean gray value is 214.43478891281822
Square for area 7  is 190380  pixels, while text area is 17104 pixels,word approximated count is 38 and mean gray value is 217.76937703540287
Square for area 8  is 805409  pixels, while text area is 80460 pixels,word approximated count is 230 and mean gray value is 214.21911476032673
Square for area 9  is 765528  pixels, while text area is 80277 pixels,word approximated count is 189 and mean gray value is 214.35945909228664
Square for area 10  is 189189  pixels, while text area is 18418 pixels,word approximated count is 35 and mean gray value is 217.8459952745667
Square for area 11  is 243729  pixels, while text area is 23033 pixels,word approximated count is 63 and mean gray value is 216.68474822446242
Square for area 12  is 2632  pixels, while text area is 22 pixels,word approximated count is 2 and mean gray value is 210.286094224924
Square for area 13  is 189810  pixels, while text area is 18339 pixels,word approximated count is 42 and mean gray value is 217.78322006216743
Square for area 14  is 5184  pixels, while text area is 518 pixels,word approximated count is 2 and mean gray value is 212.15663580246914
Square for area 15  is 46494  pixels, while text area is 5134 pixels,word approximated count is 9 and mean gray value is 213.1733341936594
Square for area 16  is 675  pixels, while text area is 18 pixels,word approximated count is 2 and mean gray value is 208.0311111111111
```

Image 50. Results for image 4_noise.png

```
Square for area 1  is 761474  pixels, while text area is 67278 pixels,word approximated count is 128 and mean gray value is 234.11333151230377
Square for area 2  is 934908  pixels, while text area is 112360 pixels,word approximated count is 182 and mean gray value is 234.05734040140848
Square for area 3  is 140840  pixels, while text area is 9513 pixels,word approximated count is 15 and mean gray value is 247.1418205055382
Square for area 4  is 661320  pixels, while text area is 65322 pixels,word approximated count is 130 and mean gray value is 234.70952035323293
Square for area 5  is 1380744  pixels, while text area is 112034 pixels,word approximated count is 338 and mean gray value is 233.54846082981348
Square for area 6  is 127635  pixels, while text area is 9340 pixels,word approximated count is 15 and mean gray value is 247.28552513025423
Square for area 7  is 871740  pixels, while text area is 86179 pixels,word approximated count is 162 and mean gray value is 233.93433592584944
Square for area 8  is 246402  pixels, while text area is 14883 pixels,word approximated count is 97 and mean gray value is 242.69369972646325
Square for area 9  is 9996  pixels, while text area is 369 pixels,word approximated count is 2 and mean gray value is 245.65376150460185
Square for area 10  is 187488  pixels, while text area is 15648 pixels,word approximated count is 26 and mean gray value is 246.04167733401604
Square for area 11  is 28261  pixels, while text area is 1530 pixels,word approximated count is 8 and mean gray value is 246.31219702062913
Square for area 12  is 6565  pixels, while text area is 720 pixels,word approximated count is 3 and mean gray value is 244.78781416603198
Square for area 13  is 44330  pixels, while text area is 5455 pixels,word approximated count is 9 and mean gray value is 247.05190615835778
```

Image 51. Results for image 5_original.png

```
Square for area 1  is 652536  pixels, while text area is 64651 pixels,word approximated count is 138 and mean gray value is 214.4488871112092
Square for area 2  is 556665  pixels, while text area is 57488 pixels,word approximated count is 135 and mean gray value is 214.74106688942183
Square for area 3  is 190380  pixels, while text area is 17126 pixels,word approximated count is 35 and mean gray value is 217.75938123752496
Square for area 4  is 388702  pixels, while text area is 36165 pixels,word approximated count is 74 and mean gray value is 215.45757675545792
Square for area 5  is 556871  pixels, while text area is 56611 pixels,word approximated count is 128 and mean gray value is 214.72619511520622
Square for area 6  is 751751  pixels, while text area is 77617 pixels,word approximated count is 164 and mean gray value is 214.36657882729787
Square for area 7  is 190380  pixels, while text area is 17255 pixels,word approximated count is 38 and mean gray value is 217.75938123752496
Square for area 8  is 805409  pixels, while text area is 81086 pixels,word approximated count is 228 and mean gray value is 214.14589481865735
Square for area 9  is 765528  pixels, while text area is 80802 pixels,word approximated count is 185 and mean gray value is 214.28917035039868
Square for area 10  is 189189  pixels, while text area is 18538 pixels,word approximated count is 35 and mean gray value is 217.83616383616385
Square for area 11  is 243729  pixels, while text area is 23274 pixels,word approximated count is 62 and mean gray value is 216.65930603251974
Square for area 12  is 2632  pixels, while text area is 22 pixels,word approximated count is 2 and mean gray value is 210.2644376899696
Square for area 13  is 189810  pixels, while text area is 18460 pixels,word approximated count is 42 and mean gray value is 217.77300985195723
Square for area 14  is 5184  pixels, while text area is 519 pixels,word approximated count is 2 and mean gray value is 212.1454475308642
Square for area 15  is 46494  pixels, while text area is 5141 pixels,word approximated count is 9 and mean gray value is 213.17348475072052
Square for area 16  is 675  pixels, while text area is 24 pixels,word approximated count is 1 and mean gray value is 207.95703703703703
```

Image 52. Results for image 5_noise.png

- ***References***
  - OpenCV Documentation: https://docs.opencv.org/3.4/index.html
  - Wikipedia: https://en.wikipedia.org/wiki/Main_Page
  - NumPy Documentation: https://numpy.org/doc/stable/index.html
  - Richard Szeliski: Computer Vision Algorithms and Applications
  - Joseph N. Wilson, Gerhard X. Ritter: Handbook of computer vision algorithms in image algebra (2001)
  - Information about morphological operations: http://homepages.inf.ed.ac.uk/rbf/HIPR2/morops.htm

Papadopoulos Aristeidis

A.M.:57576