



# Αναφορά στην 1<sup>η</sup> εργασία Όραση Υπολογιστών



Χανιώτης Παναγιώτης  
57636

17 – 11 – 2021

## Περιεχόμενα

Πρόλογος	Σελίδα 3
Θεωρητική προσέγγιση	Σελίδα 4
Πρακτικό μέρος	Σελίδα 10
Παρατηρήσεις	Σελίδα 18
Βιβλιογραφία	Σελίδα 20



## Πρόλογος

Ως άνθρωποι, αντιλαμβανόμαστε την τρισδιάστατη δομή του κόσμου γύρω μας με φαινομενική ευκολία. Ερευνητές στην όραση υπολογιστών έχουν αναπτύξει, παράλληλα, μαθηματικές τεχνικές για την ανάκτηση του τρισδιάστατου σχήματος και εμφάνισης αντικειμένων σε εικόνες. Όταν η όραση υπολογιστή ξεκίνησε για πρώτη φορά στις αρχές της δεκαετίας του 1970, θεωρήθηκε ως η συστατικό οπτικής αντίληψης μιας φιλόδοξης ατζέντας για τη μίμηση της ανθρώπινης νοημοσύνης και για να προικίζουν τα ρομπότ με έξυπνη συμπεριφορά.

Στην συγκεκριμένη αναφορά θα ασχοληθούμε με την εφαρμογή κάποιων μεθόδων της υπολογιστικής όρασης για διαμόρφωση εικόνας ώστε να αντλήσουμε κάποια χαρακτηριστικά από αυτή.

implications of building viewer or consumer profiles through data mining, within the context of VPM and in light of the reaction it already has prompted [2]. Our intent is not to answer all of the questions here, but rather to outline the issues and to propose a framework within which both academics and practitioners can further explore the issue of privacy. Our analysis centers around three essential issues involving technology and privacy: stakeholder perceptions regarding the fairness of the *process* used by a company for collecting and distributing personal information, including the level of choice provided to the individual regarding whether and to what extent they will provide access to their personal characteristics; stakeholder perceptions regarding the fairness of the *outcomes* of those processes, including the cost-benefit trade-offs inherent in the exchange of personal information for some real or perceived gain; and stakeholder perceptions regarding the *accuracy of inferred personal information*—particularly the differential perceptions of consumers and advertisers regarding

includes less salient technologies such as spyware, which is conceptually similar to the PVR in its PC-based monitoring and reporting capabilities [10]. As we note later, the main difference between the PVR and spyware is that while PVR companies attempt to communicate their data collection and usage procedures in clearly worded privacy policies, spyware often skirts ethical constraints by installing itself on a user's PC with little or no advance warning.

**T**he other major privacy issue in turn involves treatment of the information after it is collected. TiVo, a leading PVR manufacturer and service provider, is selling the viewing behaviors of its customers to Nielsen Media Research, which in turn will use that information to enhance its own collection and analysis of television viewing [6]. The type of information collected can be exceptionally detailed. TiVo was able to determine precisely who watched which commercials during the Super Bowl, the amount of time viewers spent watching commercials, and the number of times a particular viewer might have paused, rewound, and rewatched a particular segment of the game or commercial. Not surprisingly, this simple ability to collect viewing data, combined with the increasing number of PVRs in homes, has raised concerns from privacy advocates [1, 3].

The monitoring-profiling capabilities of the PVR have a direct impact on each of the stakeholders involved in the creation, distribution, and consumption of television advertising. In the domain of PVR-based targeted advertising, the direct stakeholders include television viewers, PVR providers, service providers (for example, cable and satellite companies), content providers (for example, broadcast and cable networks), and advertisers. The impact on these stakeholders can be explored, in turn, in the context of the three privacy issues raised earlier: procedures used to collect and distribute information, the perceived accuracy of those procedures, and the accuracy of inferred personal information. We discuss each of these issues, in part using TiVo as an example when appropriate, and then summarize the issues within a proposed framework for further investigation.

*TiVo was able to determine precisely who watched which commercials during the Super Bowl, the amount of time viewers spent watching commercials, and the number of times a particular viewer might have paused, rewound, and rewatched a particular segment of the game or commercial.*

the impact of accurate versus inaccurate profiles. The analytic capability of VPM adds a layer of complexity to these issues by increasing the ability of a company to gather personal (viewing) information in an exceptionally unobtrusive manner, and then to use that information to infer individual demographic characteristics.

**NEW THREATS TO PRIVACY THROUGH VIEWER PROFILING**  
The new threat to privacy begins with the basic technology of the Personal Video Recorder (PVR), which is able to directly monitor and report the viewing choices of individuals. To some extent, this is a reflection of the increased monitoring of individual behavior through a variety of data-gathering means, including point-of-sale devices, online ordering forms, and product registration requests. It also

120 May 2005/VOL. 18, NO. 3 COMMUNICATIONS OF THE ACM

implications of building viewer or consumer profiles through data mining, within the context of VPM and in light of the reaction it already has prompted [2]. Our intent is not to answer all of the questions here, but rather to outline the issues and to propose a framework within which both academics and practitioners can further explore the issue of privacy. Our analysis centers around three essential issues involving technology and privacy: stakeholder perceptions regarding the fairness of the *process* used by a company for collecting and distributing personal information, including the level of choice provided to the individual regarding whether and to what extent they will provide access to their personal characteristics; stakeholder perceptions regarding the fairness of the *outcomes* of those processes, including the cost-benefit trade-offs inherent in the exchange of personal information for some real or perceived gain; and stakeholder perceptions regarding the *accuracy of inferred personal information*—particularly the differential perceptions of consumers and advertisers regarding

includes less salient technologies such as spyware, which is conceptually similar to the PVR in its PC-based monitoring and reporting capabilities [10]. As we note later, the main difference between the PVR and spyware is that while PVR companies attempt to communicate their data collection and usage procedures in clearly worded privacy policies, spyware often skirts ethical constraints by installing itself on a user's PC with little or no advance warning.

**T**he other major privacy issue in turn involves treatment of the information after it is collected. TiVo, a leading PVR manufacturer and service provider, is selling the viewing behaviors of its customers to Nielsen Media Research, which in turn will use that information to enhance its own collection and analysis of television viewing [6]. The type of information collected can be exceptionally detailed. TiVo was able to determine precisely who watched which commercials during the Super Bowl, the amount of time viewers spent watching commercials, and the number of times a particular viewer might have paused, rewound, and rewatched a particular segment of the game or commercial. Not surprisingly, this simple ability to collect viewing data, combined with the increasing number of PVRs in homes, has raised concerns from privacy advocates [1, 3].

The monitoring-profiling capabilities of the PVR have a direct impact on each of the stakeholders involved in the creation, distribution, and consumption of television advertising. In the domain of PVR-based targeted advertising, the direct stakeholders include television viewers, PVR providers, service providers (for example, cable and satellite companies), content providers (for example, broadcast and cable networks), and advertisers. The impact on these stakeholders can be explored, in turn, in the context of the three privacy issues raised earlier: procedures used to collect and distribute information, the perceived accuracy of those procedures, and the accuracy of inferred personal information. We discuss each of these issues, in part using TiVo as an example when appropriate, and then summarize the issues within a proposed framework for further investigation.

*TiVo was able to determine precisely who watched which commercials during the Super Bowl, the amount of time viewers spent watching commercials, and the number of times a particular viewer might have paused, rewound, and rewatched a particular segment of the game or commercial.*

the impact of accurate versus inaccurate profiles. The analytic capability of VPM adds a layer of complexity to these issues by increasing the ability of a company to gather personal (viewing) information in an exceptionally unobtrusive manner, and then to use that information to infer individual demographic characteristics.

**NEW THREATS TO PRIVACY THROUGH VIEWER PROFILING**  
The new threat to privacy begins with the basic technology of the Personal Video Recorder (PVR), which is able to directly monitor and report the viewing choices of individuals. To some extent, this is a reflection of the increased monitoring of individual behavior through a variety of data-gathering means, including point-of-sale devices, online ordering forms, and product registration requests. It also

120 May 2005/VOL. 18, NO. 3 COMMUNICATIONS OF THE ACM



## Θεωρητική προσέγγιση

Ο χωρισμός μίας φωτογραφίας κειμένου σε υποπαραγράφους και η ανάλυση των επιμέρους τμημάτων της απαιτεί να ακολουθηθούν κάποια βήματα. Αρχικά, η εικόνα έχει θόρυβο τύπου 'αλατιού και πιπεριού'. Αυτός ο θόρυβος δημιουργείται από παρεμβολές στην μετάδοση του σήματος που περιέχει την πληροφορία και εμφανίζεται ως άσπρες και μαύρες κουκίδες κατά την προβολή της εικόνας.

$$\text{Grayscale} = (R + G + B) / 3.$$



Original image



Salt and pepper noise



Για την εξάλειψη αυτού του θορύβου θα πρέπει αρχικά να αλλάξουμε το χρώμα της εικόνας σε γκρι. Η τιμή του κάθε pixel γίνεται ίση την τιμή του μέσου όρου των τριών καναλιών που αντιστοιχούν στα χρώματα RGB.



Στη συνέχεια πρέπει να περάσουμε μια σειρά από φίλτρα προκειμένου να εκτελέσουμε τη διαδικασία closing η οποία είναι η πιο αποτελεσματική μέθοδος εξάλειψης αυτού του θορύβου σε δυαδικές εικόνες.



image

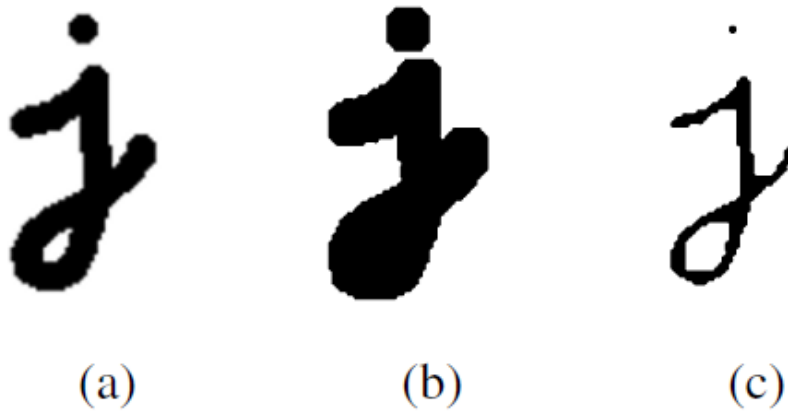
Η μετατροπή της εικόνας σε δυαδική γίνεται κανονικοποιώντας όλες τις τιμές των εικονοστοιχείων που είναι μεγαλύτερες κάποιου σε μια τιμή και αφήνοντας όλα τα υπόλοιπα στο μηδέν βάσει μιας τιμής threshold (κατώφλι) σύμφωνα και με την παρακάτω σχέση

$$\theta(f, t) = \begin{cases} 1 & \text{if } f \geq t, \\ 0 & \text{else,} \end{cases}$$

Τα δυο ενδιάμεσα βήματα που απαιτεί η συγκεκριμένη διαδικασία είναι το dilation και μετά το erosion.

- **erosion:**  $\text{erode}(f, s) = \theta(c, S)$ ;
- **majority:**  $\text{maj}(f, s) = \theta(c, S/2)$ ;
- **opening:**  $\text{open}(f, s) = \text{dilate}(\text{erode}(f, s), s)$ ;
- **closing:**  $\text{close}(f, s) = \text{erode}(\text{dilate}(f, s), s)$ .





(α) Αρχική εικόνα

(β) dilation

(γ) erosion

Όπως μπορούμε να δούμε από το σχήμα η dilation (διαστολή) μεγαλώνει αντικείμενα που αποτελούνται από 1, ενώ η erosion (διάβρωση) τα συρρικνώνει (αραιώνει). Οι εργασίες ανοίγματος και κλεισίματος τείνουν να αφήνουν μεγάλες περιοχές και λεία όρια ανεπηρέαστα, ενώ αφαιρούνται μικρά αντικείμενα ή τρύπες και λειαίνονται τα όρια.

Στο πρακτικό κομμάτι, για να γίνει η πρώτη διαδικασία πρέπει να οριστεί αρχικά ένας πίνακας με τιμές 0 και 1 του οποίου οι διαστάσεις αλλά και ο τρόπος που τα στοιχεία του είναι τοποθετημένα έχουν σημαντικό ρόλο. Αυτός ο πίνακας τον ονομάζουμε kernel (πύρινα). Αυτό ο πίνακας πολλαπλασιάζεται με τις τιμές της εικόνας και στην συνέχεια, αφού αθροίσουμε όλα τα στοιχεία του πίνακα τα τοποθετούμε στο μεσαίο pixel.

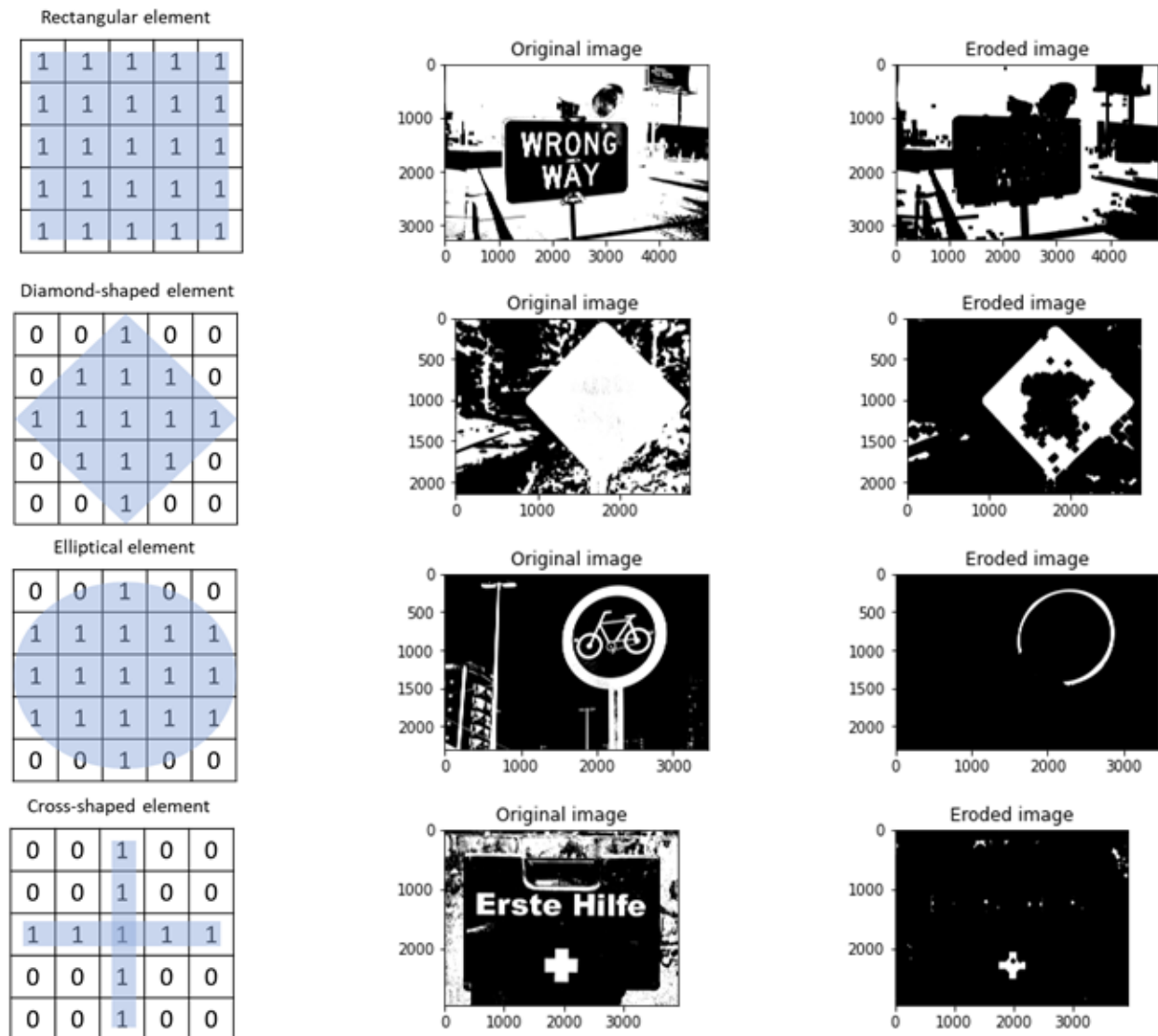
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	





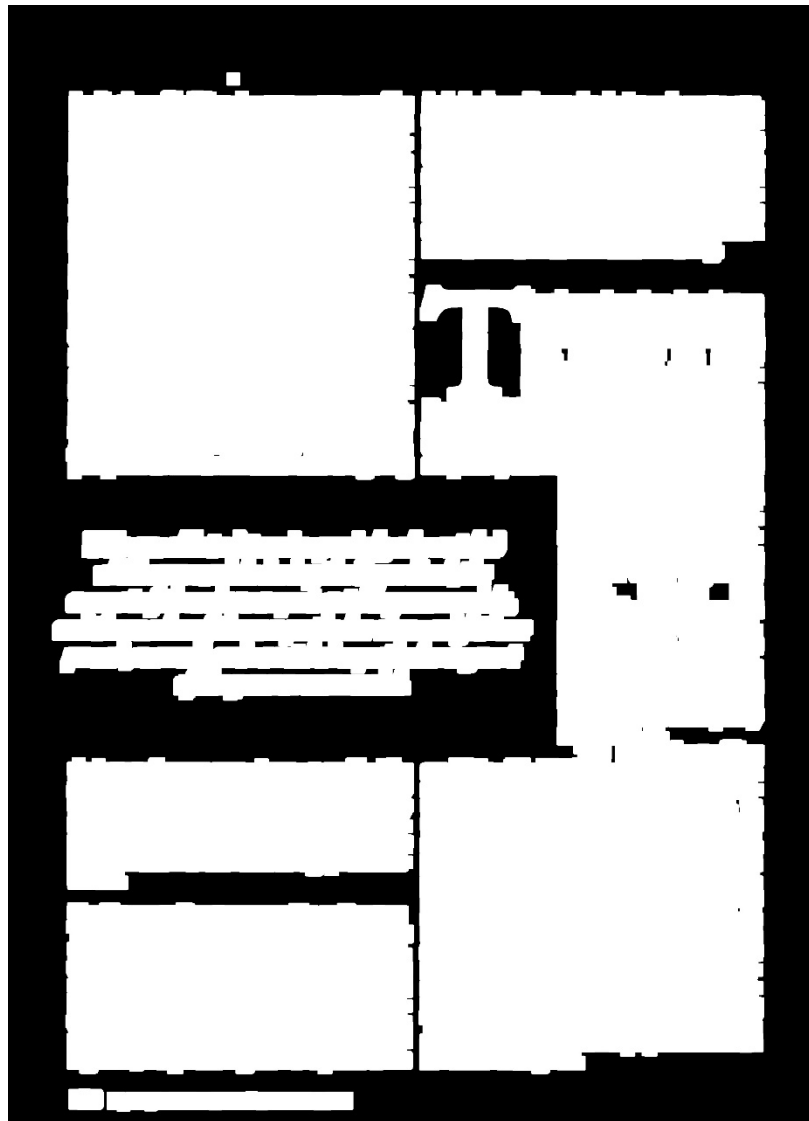
Το σχήμα του πύρινα ή του συστατικού στοιχείου, έχει σημασία όπως είπαμε παραπάνω για το αποτέλεσμα αυτής της πράξης και κατά συνέπεια του τελικού οπτικού αποτελέσματος. Στην παρακάτω εικόνα φαίνονται μερικά παραδείγματα



Η διαδικασία dilation κάνει το ακριβώς ανάποδο από την erosion. Δηλαδή μεγαλώνει τα άσπρα στοιχεία.



Μετά την αποθορυβοποίηση θα προσπαθήσουμε να ξεχωρίσουμε τις παραγράφους μεταξύ τους. Η πράξη dilation όπως είπαμε παραπάνω μεγαλώνει τα άσπρα στοιχεία σύμφωνα με τον πίνακα structuring element. Αν έχουμε ένα αρκούντως μεγάλο στοιχείο ώστε να μετατρέψει όλα τα άσπρα γράμματα μιας λέξης σε ένα αδιάκοπο σχήμα μετά το μόνο που μας μένει να κάνουμε είναι να ελέγξουμε τις περιοχές που δεν ακουμπάνε μεταξύ τους και να τις ομαδοποιήσουμε. Η υλοποίηση του παραπάνω είναι αυτή



Η εναλλαγή των χρωμάτων από μαύρο σε άσπρο έγινε διότι η συνάρτηση της OpenCV ομαδοποιεί τις άσπρες περιοχές μεταξύ τους. Θα αναληθώ αναλυτικά στο κομμάτι της υλοποίησης





Αφού ομαδοποιήσουμε τις παραγράφους, αποκτήσουμε δεδομένα για την θέση τους και χαράξουμε τα κουτιά που τις περικλείουν θα μπορούμε στην διαδικασία να τμηματίσουμε την εικόνα σε μικρότερες όπου η κάθε μια θα έχει μέσα της μόνο την εκάστοτε παράγραφο. Εκεί θα κάνουμε για άλλη μια φορά μορφολογικές διεργασίες προκειμένου να ξεχωρίσουμε την κάθε λέξη.

Ο χωρισμός των λέξεων γίνεται με την ίδια διαδικασία που χωρίσαμε τις παραγράφους, βέβαια αυτή τη φορά θα χρησιμοποιηθεί μικρότερος πίνακας για να μην ενοποιηθούν όλες οι λέξεις μεταξύ τους αλλά μόνο τα γράμματα αυτών των λέξεων. Αφού ομαδοποιηθούν και οι λέξεις, μπορούμε να τις μετρήσουμε και στη συνέχεια γνωρίζοντας τις διαστάσεις του κουτιού που περικλείνει τις παραγράφους να υπολογίσουμε το εμβαδό αλλά και το μέσο όρο της κλίμακας του γκρι.

**implications of building viewer or consumer profiles through data mining, within the context of VPM and in light of the reaction it already has prompted [2]. Our intent is not to answer all of the questions here, but rather to outline the issues and to propose a framework within which both academics and practitioners can further explore the issue of privacy. Our analysis centers around three essential issues involving technology and privacy: stakeholder perceptions regarding the fairness of the *process* used by a company for collecting and distributing personal information, including the level of choice provided to the individual regarding whether and to what extent they will provide access to their personal characteristics; stakeholder perceptions regarding the fairness of the *outcomes* of those processes, including the cost-benefit trade-offs inherent in the exchange of personal information for some real or perceived gain; and stakeholder perceptions regarding the *accuracy of inferred personal information*—particularly the differential perceptions of consumers and advertisers regarding**

from

example

the

Περισσότερες εικόνες από υποπαραγράφους και λέξεις υπάρχουν στους αντίστοιχους φακέλους της αναφοράς



## Πρακτικό μέρος

Για την υλοποίηση του κώδικα θα χρειαστούμε συναρτήσεις και κλάσεις από τις βιβλιοθήκες cv2 και numpy αντίστοιχα. Η πρώτη μας προσφέρει όλα τα εργαλεία για εισαγωγή, εξαγωγή, επεξεργασία και αναγνώριση εικόνων. Η βιβλιοθήκη numpy, μιμούμενη τις αντίστοιχες εντολές της γλώσσας του προγράμματος MATLAB μας προσφέρει εργαλεία για πράξεις με πίνακες και άλλες δομές δεδομένων με πολλές διαστάσεις.

```
import cv2
import numpy as np
```

Για την ανάγνωση των εικόνων χρησιμοποιούμε την παρακάτω εντολή. Τις διαβάζουμε κατευθείαν ως ασπρόμαυρες για τους λόγους που εξηγήσαμε στην θεωρία. Η εντολή επανάληψης μας επιτρέπει να διαβάσουμε και τις 5 ζητούμενες εικόνες από τον φάκελο images. Μπορούμε μάλιστα να επιλέξουμε αν θα αναγνώσουμε αυτές με θόρυβο ή αυτές χωρίς.

```
for a in range(1,6):
    #import photo
    filename = 'images/'+str(a)+'_'+noise_type+'.png'
    img = cv2.imread(filename, cv2.IMREAD_GRAYSCALE)
```

Στη συνέχεια θα κάνουμε αποθορυβοποίηση ακολουθώντας την μεθοδολογία closing. Στην OpenCV αυτό γίνεται πολύ γρήγορα με την παρακάτω εντολή

```
#CLOSING WITH OPEN CV

# den_ker = 3
# strel = cv2.getStructuringElement(cv2.MORPH_CROSS, (den_ker,den_ker))
# img = cv2.morphologyEx(img, cv2.MORPH_CLOSE, strel)
```

Λόγω των αναγκών της παρούσας εργασίας θα γράψουμε μια δική μας υλοποίηση.



Αρχικά θα αντιγράψουμε προσωρινά την εικόνα στις μεταβλητές `img_temp` και `img2` θα επιλέξουμε το συστατικό μας στοιχείο (`strel`) ή αλλιώς τον πύρινα μας (`kernel`) για να εκτελέσουμε το `dilation`. Ο πίνακας μας θα είναι μεγέθους 3 και θα έχει το σχήμα ενός σταυρού. Θα μπορούσε να οριστεί ως πίνακας `numpy` χωρίς την εντολή `getStructuringElement` αλλά αυτό θα ήταν μη πρακτικό γιατί αν θέλαμε να αλλάξουμε τις διαστάσεις του πίνακα γρήγορα θα έπρεπε να εισάγουμε νέα στοιχεία χειροκίνητα.

Κρατάμε τις διαστάσεις της εικόνας στις εντολές `H`, `W` για ύψος και πλάτος αντίστοιχα και μετά ορίζουμε ένα `pad` που θα περιβάλει την εικόνα και θα μας διαβεβαιώσει ότι κατά τη διάρκεια της πράξης `cross-correlation` δεν θα βγούμε εκτός ορίων της.

Όπως αναφέρθηκε στη θεωρία, αυτές οι μέθοδοι λειτουργούν μόνο σε δυαδικές εικόνες, οπότε επιλέγουμε ένα `threshold` και επιβάλλουμε κάθε τιμή που είναι πάνω από 210 να γίνει 255 (άσπρο). και όλες οι υπόλοιπες να γίνουν 0 (μαύρο).

(η εντολή επιστρέφει σαν έξοδο δυο τιμές η τιμή `[0]` μας επιστρέφει το `threshold`, επομένως δεν μας ενδιαφέρει και ζητάμε μόνο την τιμή `[1]` που περιέχει την αλλαγμένη εικόνα.

```
img_temp = img
img2 = img
# dilation kernel size
den_ker = 3
kernel = cv2.getStructuringElement(cv2.MORPH_CROSS, (den_ker, den_ker))
H, W = img.shape
pad = int((kernel.shape[1] - 1) / 2)
img_temp = cv2.copyMakeBorder(img_temp, pad, pad, pad, pad, cv2.BORDER_DEFAULT)
cv2.destroyWindow('main')

# threshold
img_temp = cv2.threshold(img_temp, 210, 255, cv2.THRESH_BINARY)[1]
```

Η διαδικασία `dilation`, όπως περιγράψαμε θα γίνει με πολλαπλασιασμούς περιοχών στο μέγεθος του πύρινα (`matrix`) μας με τον πίνακα του πύρινα (`kernel`). Το αποτέλεσμα της πράξης του πολλαπλασιασμού μας δίνει έναν νέο πίνακα του οποίου αφού αθροίσουμε όλα τα στοιχεία, αποθηκεύουμε σε μια μεταβλητή `sum`. Η τιμή της θα αποδοθεί στο μεσαίο pixel αν είναι 0 αλλιώς αυτό θα λάβει την τιμή 255.

```
# DILATION
for x in np.arange(pad, H + pad):
    for y in np.arange(pad, W + pad):
        matrix = img_temp[x - pad:x + pad + 1, y - pad:y + pad + 1]
        sum = np.sum(matrix * kernel)
        if sum == 0:
            pixel = 0
        else:
            pixel = 255
        img2[x - pad, y - pad] = pixel
```



Η ίδια διαδικασία θα γίνει για το erosion, με ουσιαστική διαφορά το μέγεθος του πίνακα kernel το οποίο θα είναι μεγαλύτερο (5x5) αλλά και την συνθήκη σύμφωνα με την οποία θα λάβει τιμή το κεντρικό pixel, η οποία θα είναι η αντίστροφη.

```
# erosion kernel size
den_ker = 5
kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (den_ker, den_ker))
kernel_sum = np.sum(kernel)
img3 = img2
pad = int((kernel.shape[1] - 1) / 2)
img2 = cv2.copyMakeBorder(img2, pad, pad, pad, pad, cv2.BORDER_DEFAULT)

# EROSION
for x in np.arange(pad, H + pad):
    for y in np.arange(pad, W + pad):
        matrix = img2[x - pad:x + pad + 1, y - pad:y + pad + 1]
        sum = np.sum(matrix * kernel) / kernel_sum
        if sum == 255:
            pixel = 255
        else:
            pixel = 0
        img[x - pad, y - pad] = pixel
```

Επειδή είναι ζητούμενο να περιγράψουμε τις περιοχές τις αποθρομβωποιημένης εικόνας με έγχρωμα πλαίσια θα την ορίσουμε ως έγχρωμη για να επιτραπεί αυτό.

```
# return to rgb to have colored bounding rectangle
img = cv2.cvtColor(img, cv2.COLOR_RGB2BGRA)
```

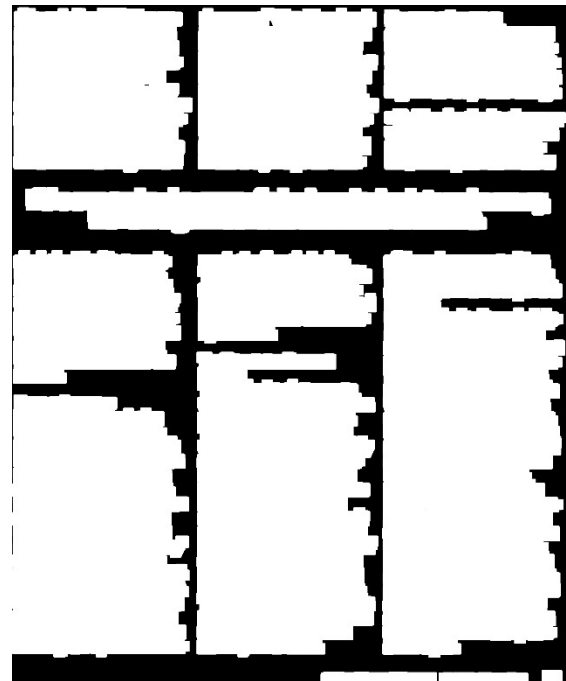
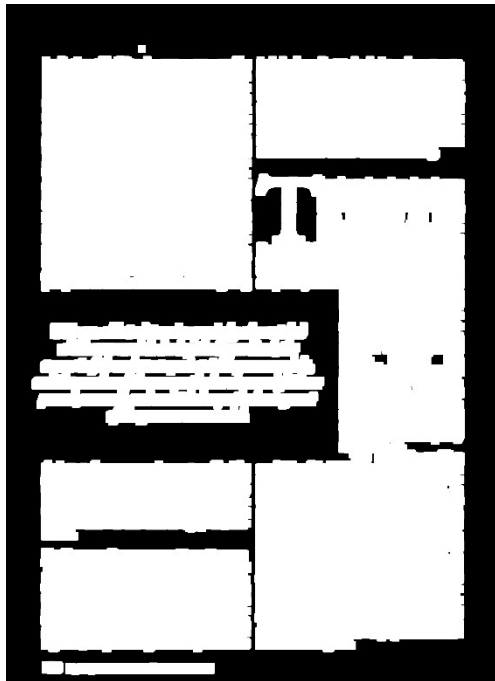
Για τον χωρισμό σε παραγράφους, όπως περιγράψαμε στην θεωρία, θα κάνουμε την πράξη erosion προκειμένου τα μέχρι τώρα μαύρα γράμματα να μεγαλώσουν σε μέγεθος και να ακουμπήσουν μεταξύ τους. Ο πυρήνας μας σε αυτή την περίπτωση είναι πολύ μεγάλος (35x35). Εδώ χρησιμοποιούνται η έτοιμη συνάρτηση morphologyEx της OpenCV για να εκτελείται ο κώδικας με μεγαλύτερη ταχύτητα.

```
#make one cluster
par_ker = 29+6
strel = cv2.getStructuringElement(cv2.MORPH_RECT, (par_ker, par_ker))
img_blob = cv2.morphologyEx(img, cv2.MORPH_ERODE, strel)
```



Στη συνέχεια θα μετατρέψουμε την ένα αντίγραφο της εικόνας σε δυαδική σύμφωνα με το ίδιο κατώφλι που χρησιμοποιήσαμε στην αποθορυβοποίηση. Αυτό γίνεται επειδή η συνάρτηση που θα δούμε παρακάτω, η `cv2.findContours` θεωρεί πάντα ως background το μαύρο και ως περιοχές τα άσπρα μέρη.

```
#make binary % invert  
thresh = 210  
binary = cv2.threshold(img_blob, thresh, 255, cv2.THRESH_BINARY_INV)[1] #this function outputs 2 values, [0] is the threshold set and [1]  
cv2.imwrite('binaries/binary' + str(a) + '.jpg', binary)
```



Στη συνέχεια βρίσκουμε τα contours (περιγράμματα). Τα περιγράμματα μπορούν να εξηγηθούν απλώς ως μια καμπύλη που ενώνει όλα τα συνεχή σημεία (κατά μήκος του ορίου), έχοντας το ίδιο χρώμα ή ένταση. Τα περιγράμματα είναι ένα χρήσιμο εργαλείο για την ανάλυση σχήματος και τον εντοπισμό και την αναγνώριση αντικειμένων.

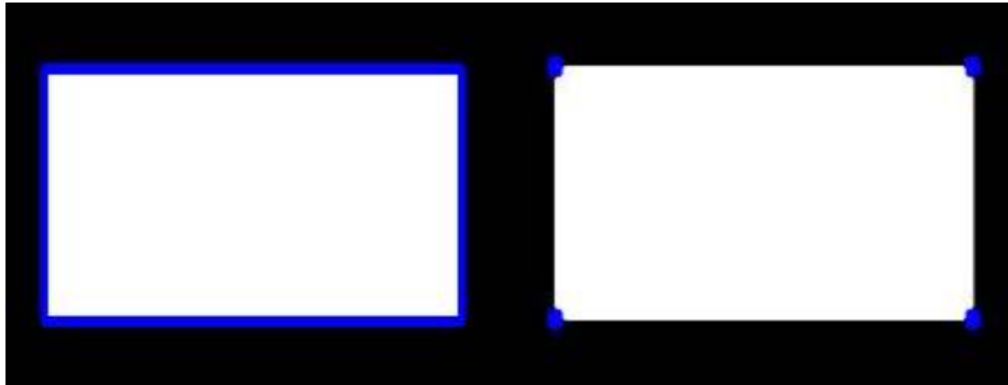
Η εντολή για να βρούμε αυτά τα σημεία είναι η

```
#find contours  
contours = cv2.findContours(binary, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)[1]
```

Η συνάρτηση έχει 3 εξόδους από τις οποίες μας ενδιαφέρει μόνο η δεύτερη [1], δηλαδή η λίστα με τις θέσεις όλων των υποπεριοχών



Τα ορίσματα RETR\_EXTERNAL μας δίνουν μόνο τα εξωτερικά περιγράμματα των αντικειμένων και το CHAIN\_APPROX\_SIMPLE μας δίνει αρκετές πληροφορίες ώστε να καταλάβουμε πλήρως τις θέσεις των στοιχείων χωρίς να δίνει περιττά δεδομένα που καταναλώνουν χώρο στη μνήμη



image

APROXIMATION\_NONE

APROXIMATION\_SIMPLE

Η συνάρτηση boundingRect μας επιστρέφει την θέση σε συντεταγμένες x,y καθώς και τα γεωμετρικά χαρακτηριστικά πλάτους και ύψους ενός κουτιού που περικλείει το contour της.

```
x,y,w,h = cv2.boundingRect(contours[i])
```

Ο αριθμός των contours είναι ο αριθμός των παραγράφων μιας σελίδας.

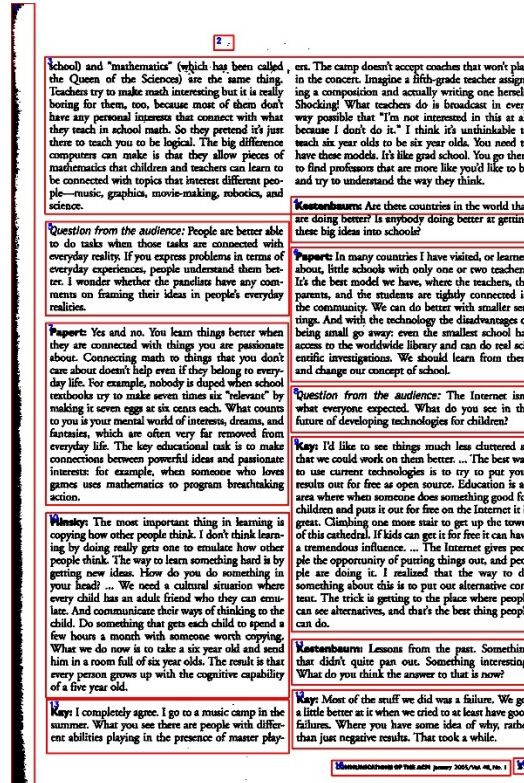


Τα contours μιας εικόνας η οποία δεν έχει περάσει τη διαδικασία erosion. Βλέπουμε ότι αναγνωρίζει ως υποπεριοχή το κάθε γράμμα.





Κάποια μικρά κομμάτια θορύβου που δεν καθαρίστηκαν επαρκώς λόγω του μεγάλου μεγέθους τους όπως βλέπουμε στην εικόνα 4



Για να αποκλείσουμε ότι αυτά τα κομμάτια θα θεωρηθούν από το πρόγραμμα ως κάποια πιθανή παράγραφος θα αναγκαστούμε να θέσουμε μια ελάχιστη τιμή εμβαδού του περιγράμματος μας ώστε να αυτό να θεωρηθεί περιοχή. Ο κώδικας αυτού είναι ο παρακάτω

```
#remove small contours
min_area = 2500
length = len(contours)
i=0
while(i<length):
    x,y,w, h = cv2.boundingRect(contours[i])
    area= w*h
    if area ≤ min_area:
        del contours[i]
        i=i-1
        length = length -1

    i = i+1
```





Μετά, ορίζουμε έναν πίνακα στον οποίο θα αποθηκεύσουμε τις παραγράφους ως εικόνες και επαναλαμβάνουμε την εξής διαδικασία για κάθε contour.

Πρώτα βρίσκουμε τις συντεταγμένες του πλαισίου γύρω από το περίγραμμα μας. Μετά, γνωρίζοντας τις συντεταγμένες τους, αποθηκεύουμε ξεχωριστά κάθε παράγραφο. Ύστερα, αποθηκεύουμε το αρχείο σε ένα φάκελο και τέλος σχεδιάζουμε τα τα κουτιά και γράφουμε τους αριθμούς με τις εντολές `cv2.rectangle` και `cv2.putText` (το κείμενο τοποθετείται με ένα offset  $+(10,30)$  για να φαίνονται καλύτερα).

```
#draw box
img_par = [] #store paragraphs here
for i in range(0,len(contours)):
    x,y,w,h = cv2.boundingRect(contours[i])

    #cut into different images
    img_temp = img[y:y + h, x:x + w]
    # cut into paragraphs
    img_par.append(img_temp)
    cv2.imwrite('img_temp'+str(a)+'/'+'temp' + str(i) + '.jpg', img_temp)

#draw boxes
cv2.rectangle(img_color, (x, y), (x + w, y + h), (0,0,255), 5)

#write numbers
img_color = cv2.putText(img_color, str(len(contours) - i), (x+10,y+30), cv2.FONT_HERSHEY_SIMPLEX,1, (255,0,0),5)

#output file
cv2.imwrite('end_result/bounding_box'+str(a)+'.'+'jpg',img_color)
```

Τελευταίος στόχος μας είναι να τυπώσουμε τα παρακάτω χαρακτηριστικά στην κονσόλα.

Αρχικά, αντιστρέφουμε τον πίνακα `img_par`. Ο λόγος που γίνεται αυτό είναι ο ίδιος λόγος για τον οποίο κατά την γραφή των αριθμών τους γράφαμε από το τέλος προς την αρχή (`str(len(contours) - i)`). Αυτό οφείλεται στην σειρά που αποθηκεύει η εντολή `getContours`, η οποία είναι από κάτω προς πάνω.

Η διαδικασία για την μέτρηση των λέξεων είναι παρόμοια με αυτή της μέτρησης παραγράφων. Η ουσιαστική διαφορά είναι ο μικρότερος `kernel` ώστε να ενωθούν μόνο τα γράμματα μιας λέξης και όχι με άλλες. Η άλλη διαφορά είναι ο τρόπος που έχουν τοποθετηθεί να 1 στον πίνακα σχηματίζοντας έλλειψη αντί για τετράγωνο που χρησιμοποιούσαμε ως τώρα (`cv2.MORPH_ELLIPSE`).

Με τα γεωμετρικά χαρακτηριστικά που μας δίνει από τις διαστάσεις των μικρότερων εικόνων υπολογίζουμε το εμβαδό και τέλος, αθροίζοντας όλες τις τιμές των `pixel` τις εικόνες και διαιρώντας τις με το εμβαδό, που αποτελεί και τον αριθμό όλων των `pixel` έχουμε και το μέσο όρο της κλίμακας του γκρι.



```
img_par.reverse();
for i in range(0,len(img_par)):

    # make one blob
    word_ker = 11
    strel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (word_ker, word_ker))
    img_blob = cv2.morphologyEx(img_par[i], cv2.MORPH_ERODE, strel)

    # make binary % invert
    thresh = 210
    img_bin = cv2.threshold(img_blob, thresh, 255, cv2.THRESH_BINARY_INV)[1]

    # find contours
    contours = cv2.findContours(img_bin, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)[1]
    print("-----REGION " + str(i + 1)+"-----")
    print("Word count #"+str(len(contours)))
    area = img_par[i].shape[1]*img_par[i].shape[0]
    print("AREA is " + str(area) + " pixels")
    print("MEAN GRAY LEVEL is " + str(np.sum(img_par[i])/area))
```

## Έξοδοι

με θόρυβο

χωρίς θόρυβο

έξοδος κονσόλας

Implications of building viewer or consumer profiles through data mining, within the context of VPM and in light of the reaction it already has prompted [1]. Our interest is not to answer all of the questions here, but rather to outline the issues and to propose a framework within which both academics and practitioners can further explore the issue of privacy. Our analysis centers around three essential issues involving technology and privacy: stakeholder perceptions regarding the fairness of the process used by a company for collecting and distributing personal information, including the level of choice provided to the individual regarding whether and to what extent they will provide access to their personal characteristics; stakeholder perceptions regarding the fairness of the outcome of those processes, including the cost-benefit trade-offs inherent in the exchange of personal information for some real or perceived gain; and stakeholder perceptions regarding the accuracy of inferred personal information—particularly the differential perceptions of consumers and advertisers regarding

**T**he other major privacy issue in turn involves treatment of the information after it is collected. Tivo, a leading PVR manufacturer and service provider, is selling the viewing behavior of its customers to Nielsen Media Research, which in turn will use that information to enhance its own collection and analysis of television viewing [6]. The type of information collected can be exceptionally detailed. Tivo was able to determine precisely who watched which commercials during the Super Bowl, the amount of time viewers spent watching commercials, and the number of times a particular viewer might have paused, rewound, and watched a particular segment of the game or commercial.

The impact of accurate versus inaccurate profiles. The analytic capability of VPM adds a layer of complexity to these issues by increasing the ability of a company to gather personal (viewing) information in an exceptionally unobtrusive manner, and then to use that information to infer individual demographic characteristics.

**New Threats to Privacy through Viewer Profiling.** The new threat to privacy begins with the basic technology of the Personal Video Recorder (PVR), which is able to directly monitor and report the viewing choices of individuals. In some extent, this is a reflection of the increased monitoring of individual behavior through a variety of data-gathering means, including point-of-sale devices, online ordering forms, and product registration requests. It also

capabilities of the PVR have a direct impact on each of the stakeholders involved in the creation, distribution, and consumption of television advertising. In the domain of PVR-based targeted advertising, the direct stakeholders include television viewers, PVR providers, service providers (for example, cable and satellite companies), content providers (for example, broadcast and cable networks), and advertisers. The impact on these stakeholders involves the perceived outcomes of these processes, and the accuracy of inferred personal information. We discuss each of these issues in part using Tivo as an example when appropriate, and then summarize the issues within a proposed framework for further investigation.

Implications of building viewer or consumer profiles through data mining, within the context of VPM and in light of the reaction it already has prompted [1]. Our interest is not to answer all of the questions here, but rather to outline the issues and to propose a framework within which both academics and practitioners can further explore the issue of privacy. Our analysis centers around three essential issues involving technology and privacy: stakeholder perceptions regarding the fairness of the process used by a company for collecting and distributing personal information, including the level of choice provided to the individual regarding whether and to what extent they will provide access to their personal characteristics; stakeholder perceptions regarding the fairness of the outcome of those processes, including the cost-benefit trade-offs inherent in the exchange of personal information for some real or perceived gain; and stakeholder perceptions regarding the accuracy of inferred personal information—particularly the differential perceptions of consumers and advertisers regarding

**T**ivo was able to determine precisely who watched which commercials during the Super Bowl, the amount of time viewers spent watching commercials, and the number of times a particular viewer might have paused, rewound, and watched a particular segment of the game or commercial.

The impact of accurate versus inaccurate profiles. The analytic capability of VPM adds a layer of complexity to these issues by increasing the ability of a company to gather personal (viewing) information in an exceptionally unobtrusive manner, and then to use that information to infer individual demographic characteristics.

**New Threats to Privacy through Viewer Profiling.** The new threat to privacy begins with the basic technology of the Personal Video Recorder (PVR), which is able to directly monitor and report the viewing choices of individuals. In some extent, this is a reflection of the increased monitoring of individual behavior through a variety of data-gathering means, including point-of-sale devices, online ordering forms, and product registration requests. It also

The other major privacy issue in turn involves treatment of the information after it is collected. Tivo, a leading PVR manufacturer and service provider, is selling the viewing behavior of its customers to Nielsen Media Research, which in turn will use that information to enhance its own collection and analysis of television viewing [6]. The type of information collected can be exceptionally detailed. Tivo was able to determine precisely who watched which commercials during the Super Bowl, the amount of time viewers spent watching commercials, and the number of times a particular viewer might have paused, rewound, and watched a particular segment of the game or commercial. No, surprisingly, this simple ability to collect viewing data combined with the increasing number of PVRs in homes, has raised concerns from privacy advocates [1, 3].

```
-----REGION 1 -----
Word count #136
AREA is 1141536 pixels
MEAN GRAY LEVEL is 222.66835123903232
-----REGION 2 -----
Word count #66
AREA is 501501 pixels
MEAN GRAY LEVEL is 223.33659753420233
-----REGION 3 -----
Word count #248
AREA is 2298435 pixels
MEAN GRAY LEVEL is 223.97786058774776
-----REGION 4 -----
Word count #59
AREA is 692416 pixels
MEAN GRAY LEVEL is 230.6270291269988
-----REGION 5 -----
Word count #44
AREA is 417312 pixels
MEAN GRAY LEVEL is 225.28403688367456
```



## Παρατηρήσεις

- 1) Η μέθοδος για τον καθαρισμό της εικόνας δεν έχει βελτιστοποιήσεις (optimizations) σαν κι αυτές που έχουν οι έτοιμες συναρτήσεις της OpenCV με αποτέλεσμα ο χρόνος εκτέλεσης να αυξάνεται δραματικά. Αυτό συμβαίνει επειδή μια πράξη cross correlation όπως είναι το dilation και το erosion αποτελείται από υπερβολικά πολλούς πολλαπλασιασμούς και συνθήκες ελέγχου με αποτέλεσμα να καταναλώνονται υπερβολικά πολλοί πόροι από έναν υπολογιστή που το εκτελεί.

Αποτελέσματα από την `time.time()`. Το πρώτο πρόγραμμα πήρε συνολικά 6 λεπτά ενώ το δεύτερο μόλις 4 δευτερόλεπτα.

```
AREA is 5240 pixels
MEAN GRAY LEVEL is 209.0825076219512
-----REGION 9 -----
Word count #2
AREA is 46935 pixels
MEAN GRAY LEVEL is 194.47043783956536
time for execution 1637167581.6462786

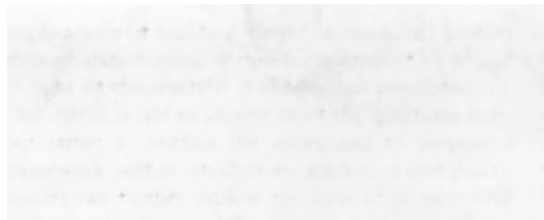
Process finished with exit code 0
```

```
AREA is 4700 pixels
MEAN GRAY LEVEL is 206.44576612903225
-----REGION 8 -----
Word count #4
AREA is 45323 pixels
MEAN GRAY LEVEL is 204.6674977384551
time for execution 1637167877.7484374

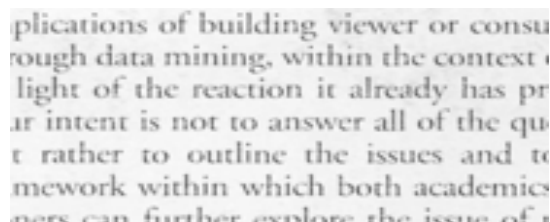
Process finished with exit code 0
```

- 2) Η μέθοδος αποθορυβοποίησης που χρησιμοποιήθηκε, ενώ είναι η καλύτερη που θα μπορούσε να χρησιμοποιηθεί στην προκειμένη περίπτωση, δεν μπορεί να διώξει επιτυχώς θόρυβο όπως αυτός στην εικόνα 4 και ο λόγος είναι ότι ο θόρυβος είναι τόσο μεγάλος που αν γίνει προσπάθεια να εξαλειφθεί με με μεγαλύτερους πίνακες στο dilation και erosion η εικόνα θα έχανε σημαντική πληροφορία και τα γράμματα δεν θα διαβάζονταν.

με μέγεθος kernel = 20



με μέγεθος kernel = 3



3) Κάποιες περιοχές εισέρχονται μέσα σε άλλες όπως για παράδειγμα στην εικόνα 1. Αυτό δημιουργεί προβλήματα στην μέτρηση λέξεων αλλά και εμβαδού υποπαραγράφων. Αυτό θα μπορούσε να λυθεί αν παίρναμε αναλυτικότερα το περίγραμμα των εικόνων με εντολές που υπάρχουν μέσα στην OpenCV. Αυτό έκανε πιο δύσκολο το σπάσιμο της εικόνας σε μικρότερες εικόνες



image

```
1 epsilon = 0.1*cv2.arcLength(cnt,True)
2 approx = cv2.approxPolyDP(cnt,epsilon,True)
```



## Βιβλιογραφία

Computer Vision Algorithms and Applications, Richard Szeliski

Digital Image Processing, Jayaraman

Computer Vision Lab Course, Lazaros Tsochatzidis

[https://docs.opencv.org/3.4/d3/df2/tutorial\\_py\\_basic\\_ops.html](https://docs.opencv.org/3.4/d3/df2/tutorial_py_basic_ops.html)

[https://docs.opencv.org/3.4/d7/d4d/tutorial\\_py\\_thresholding.html](https://docs.opencv.org/3.4/d7/d4d/tutorial_py_thresholding.html)

[https://docs.opencv.org/3.4/d4/d73/tutorial\\_py\\_contours\\_begin.html](https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html)

[https://docs.opencv.org/3.1.0/dd/d49/tutorial\\_py\\_contour\\_features.html](https://docs.opencv.org/3.1.0/dd/d49/tutorial_py_contour_features.html)

<https://www.dynamsoft.com/blog/insights/image-processing/image-processing-101-color-space-conversion/>

