|  |  |  |
| --- | --- | --- |
| **ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**  Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών | **ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ**  **ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ**  **ΑΚΑΔΗΜΑΪΚΟΥ ΕΤΟΥΣ 2022-23**  **Τεχνοοικονομικά Συστήματα**  **(Techno-economic Systems)** | **ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ**  Σχολή Ναυτιλίας & Βιομηχανίας του Τμήματος Βιομηχανικής Διοίκησης και Τεχνολογίας |

**Δημιουργία Εφαρμογής Ανάλυσης**

**Φωτογραφιών Κοινωνικών Δικτύων**

**μέσω της Πλατφόρμας Google Vision AI**

**ΕΡΓΑΣΙΑ**

**Γεώργιος Τσαβδαρίδης**

**(ΑΜ: 03203386)**

**Υπεύθυνος:** Ι. Αναγνωστόπουλος (janag@uth.gr)

**Project leader:** Γ. Ραζής (razis@dib.uth.gr)

Αθήνα, Δεκέμβριος 2022

**ΕΚΦΩΝΗΣΗ**

|  |
| --- |
| Τα τελευταία χρόνια έχει εμφανιστεί η τάση στα Κοινωνικά Δίκτυα (ΚΔ) της ραγδαίας αύξησης του πλήθος των διαμοιραζόμενων μηνυμάτων πολυμεσικού περιεχομένου (εικόνες, βίντεο). Αποτέλεσμα είναι η εμφάνιση ΚΔ (π.χ. Instagram) στοχευμένα αποκλειστικά και μόνο σε τέτοιου είδους μηνύματα. Είναι αρκετά συνήθης η προσπάθεια των χρηστών να χαρακτηρίσουν με διάφορες τεχνικές το περιεχόμενο αυτών των μηνυμάτων (π.χ. μέσω hashtags ή άλλων λεκτικών περιγραφών). Ωστόσο, αυτό δεν είναι επαρκές καθώς δεν υιοθετείται μία καθολική τεχνική και οι περιγραφές είναι συχνά υποκειμενικές. Σκοπός της εργασίας είναι η δημιουργία ενός συστήματος αυτόματης ανάλυσης και χαρακτηρισμού/επισήμανσης εικόνων ΚΔ μέσω της πλατφόρμας Google Vision AI. Το Google Vision API προσφέρει προ-εκπαιδευμένα μοντέλα μηχανικής μάθησης (machine learning) για άμεση χρήση σε εικόνες όπως ανάθεση ετικετών σε προκαθορισμένες κατηγορίες, εντοπισμός αντικειμένων και προσώπων, και ανάγνωση κειμένου. Συνεπώς, το σχετικό περιεχόμενο θα αντλείται από τα ΑΡΙ των ΚΔ (ή μπορούν να δοθεί υπάρχον σχετικό περιεχόμενο) και θα εμπλουτίζεται μέσω του Google Vision API. Έπειτα, οι πληροφορίες θα αποθηκεύονται με σκοπό τη μεταγενέστερη ανάκτηση και την περαιτέρω αξιοποίησή τους. Αντικείμενα ενασχόλησης: προγραμματισμός με Java ή Python, χρήση του Twitter / Instagram API, Google Vision AI ΑΡΙ. |

**ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ**

**1o BHMA**

**Δημιουργία Account στο Google Vision API και καθορισμός του API\_KEY**

**2o BHMA**

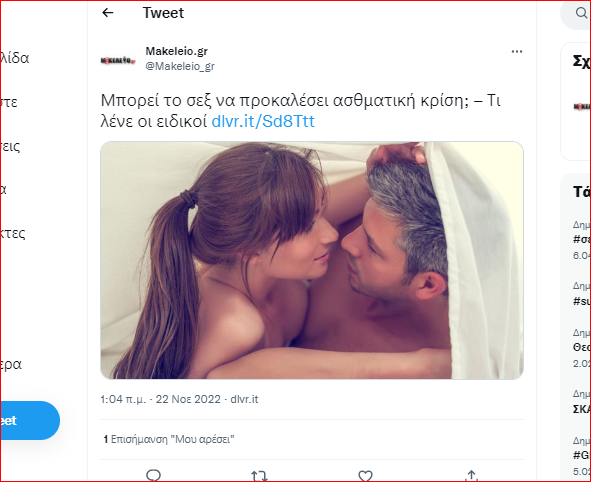
**Εύρεση Διευθύνσεων Εικόνων από Μέσα Κοινωνικής Δικτύωσης**

**Εικόνα «1»**

<https://pbs.twimg.com/media/FjQyEcCacAAK3TP?format=jpg&name=medium>

**Εικόνα «2»**



<https://pbs.twimg.com/media/FiH4qFsaYAADb9g?format=jpg&name=medium>

**Εικόνα «3»**



<https://pbs.twimg.com/media/FiIAHZ4aEAI4nRH?format=jpg&name=medium>

**Εικόνα «4»**



<https://pbs.twimg.com/media/Fiv7sp0XoAI2O9A?format=jpg&name=900x900>

**Εικόνα «5»**

<https://pbs.twimg.com/media/FiiOJKYacAAc4sJ?format=jpg&name=large>

**Εικόνα «6»**



<https://pbs.twimg.com/media/FiIdJYbWIAEkmNT?format=jpg&name=small>

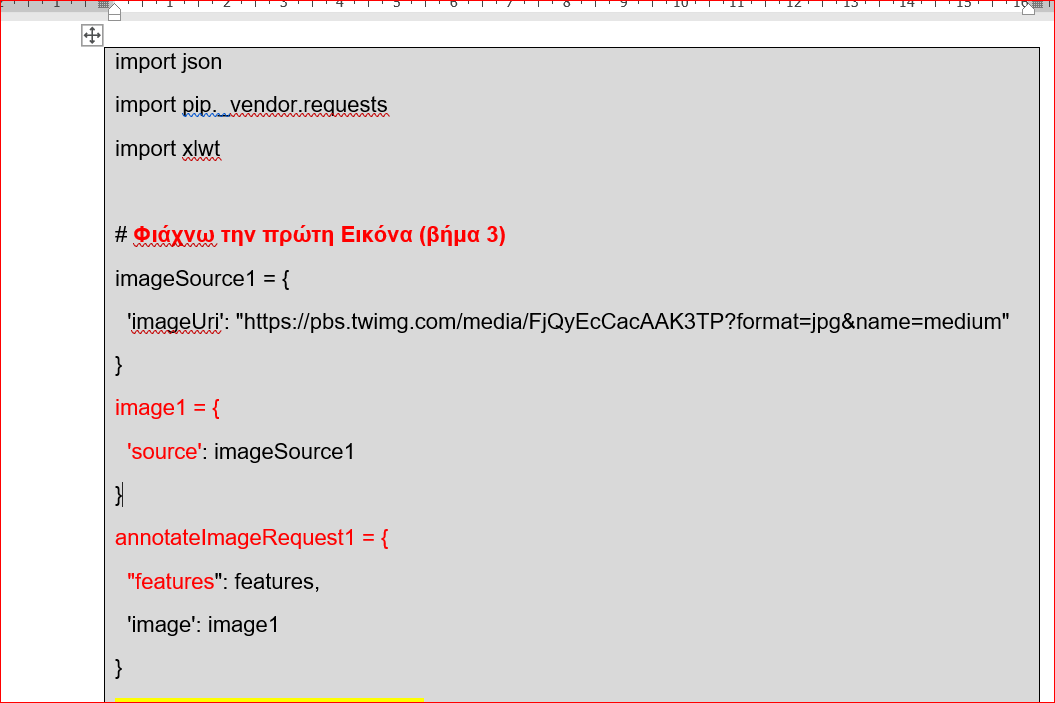
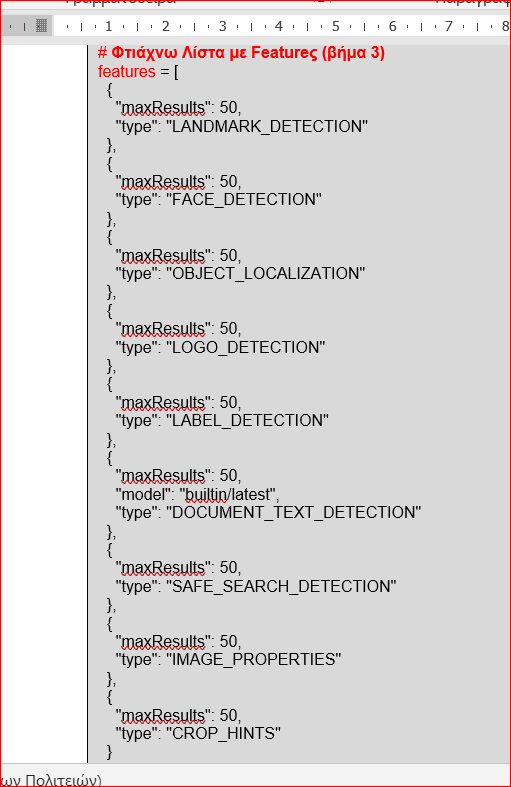
**Εικόνα «7»**



<https://pbs.twimg.com/media/FiMFwHmXwAklpOE?format=jpg&name=360x360>

**3o BHMA**

* **α. Δημιουργία Λεξικών (Dictionaries)[[1]](#footnote-1) για τις Επιλεχθείσες Εικόνες**
* **β. Δημιουργία Λιστών (Εγγραφών) με αντίστοιχα Κλειδιά (keys) και Τιμές (value)**

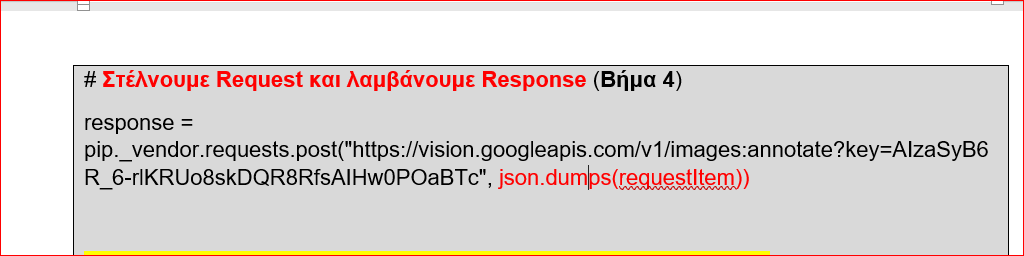


* **γ. Μέσα στον κώδικα, όμοια φτιάχνω τις λίστες για τις υπόλοιπες Εικόνες (2 έως 7)**
* **δ. Φτιάχνω το αντικείμενο Request**

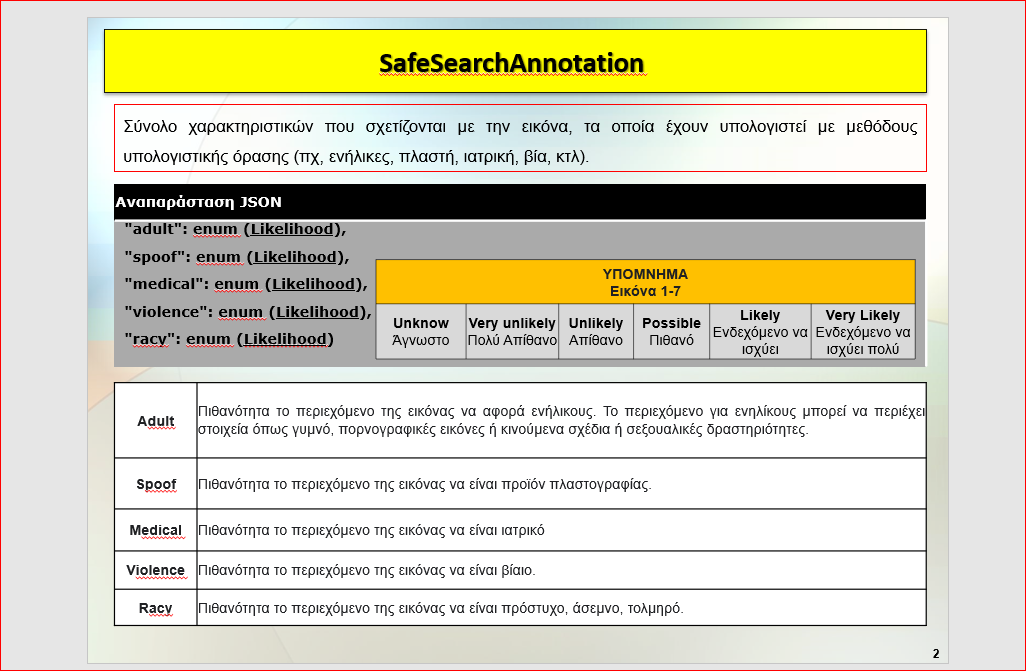


**4o BHMA**

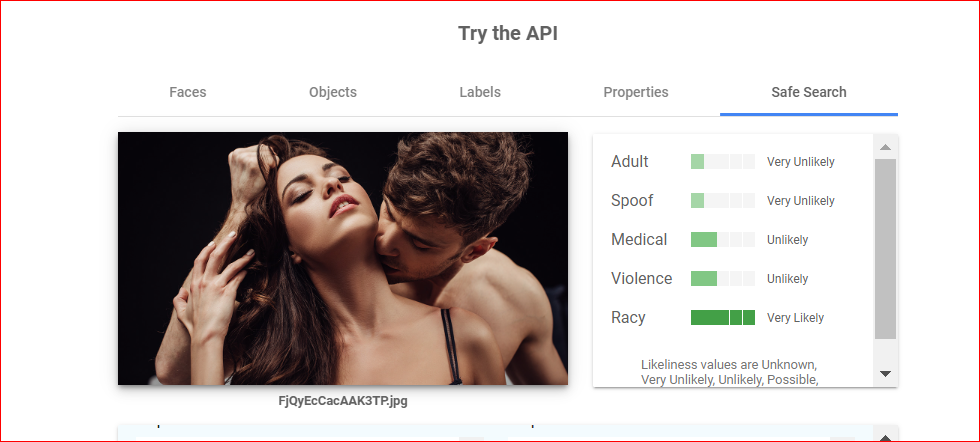
* **4α.Μετατροπή Τελικού Λεξικού** (Dictionary), **στο οποίο θα συμπεριληφθούν όλες οι οι Πληροφορίες, που περιέχουν οι Επιλεχθείσες Εικόνες σε αρχείο Json** (δομή Javascript, που χρησιμοποιείται για την επικοινωνία web εφαρμογών), **με σκοπό να σταλεί ερώτημα/αίτημα** (Request) **στο Google Vision API.**
* **4β. Αποστολή Αιτημάτων (Request) Φωτογραφιών, σε μορφή** Json, στο Google Vision API], με σκοπό να πάρω από το Google Vision API, πληροφορίες [μέθοδος Rest-API: πρωτόκολλο επικοινωνίας με Web εφαρμογές]



* **4γ. Λαμβάνω** απάντηση (response) σε μορφή Json[[2]](#footnote-2).

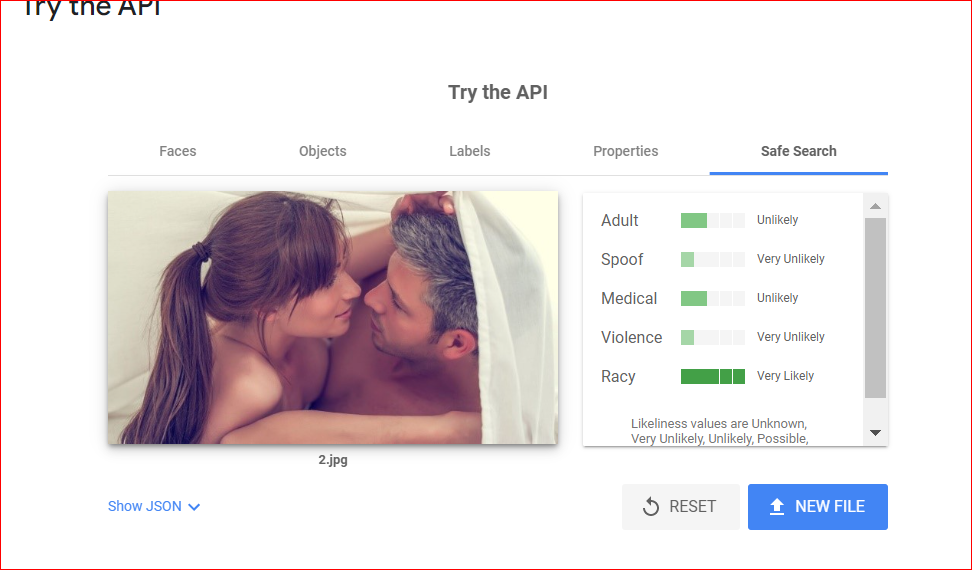


**Εικόνα «1»**



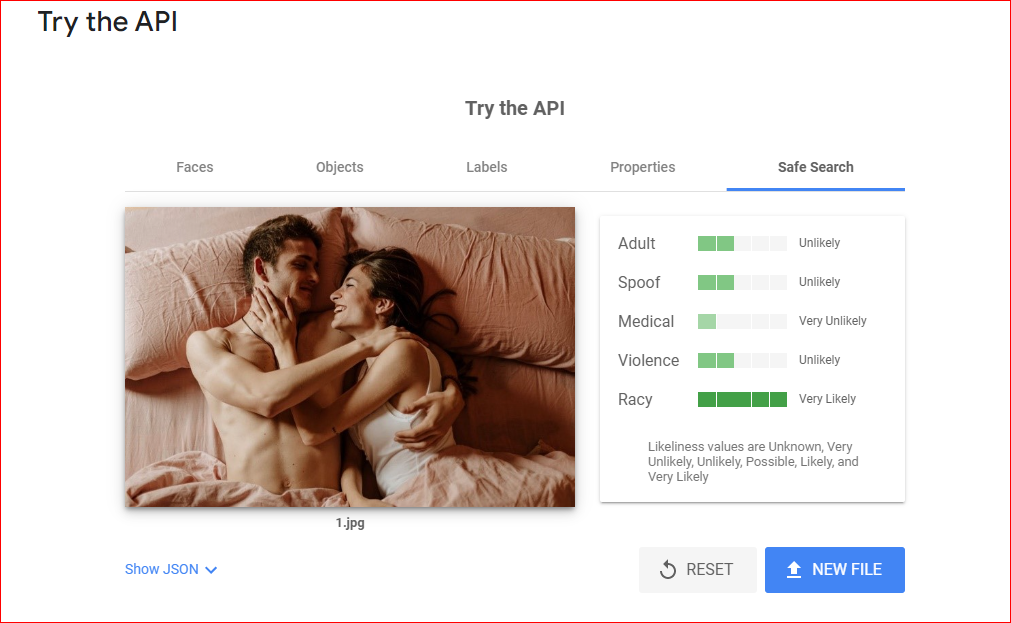
|  |
| --- |
| "safeSearchAnnotation": {  "adult": "VERY\_UNLIKELY",  "medical": "UNLIKELY",  "racy": "VERY\_LIKELY",  "spoof": "VERY\_UNLIKELY",  "violence": "UNLIKELY" |

**Εικόνα «2»**



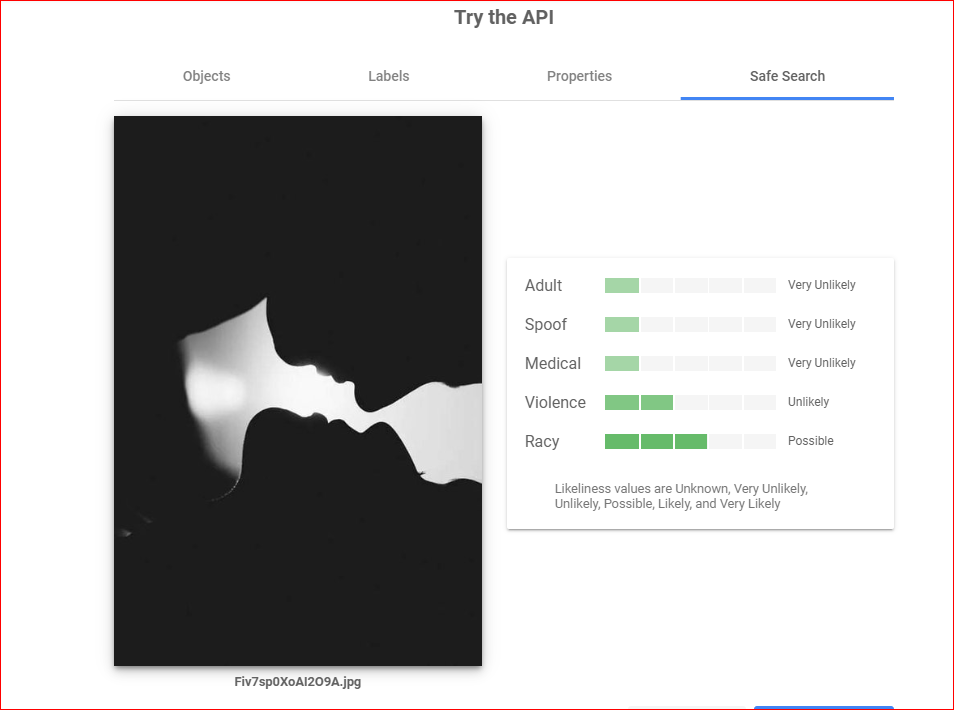
|  |
| --- |
| "safeSearchAnnotation": {  "adult": "UNLIKELY",  "medical": "UNLIKELY",  "racy": "VERY\_LIKELY",  "spoof": "VERY\_UNLIKELY",  "violence": "VERY\_UNLIKEL |

**Eικόνα «3»**



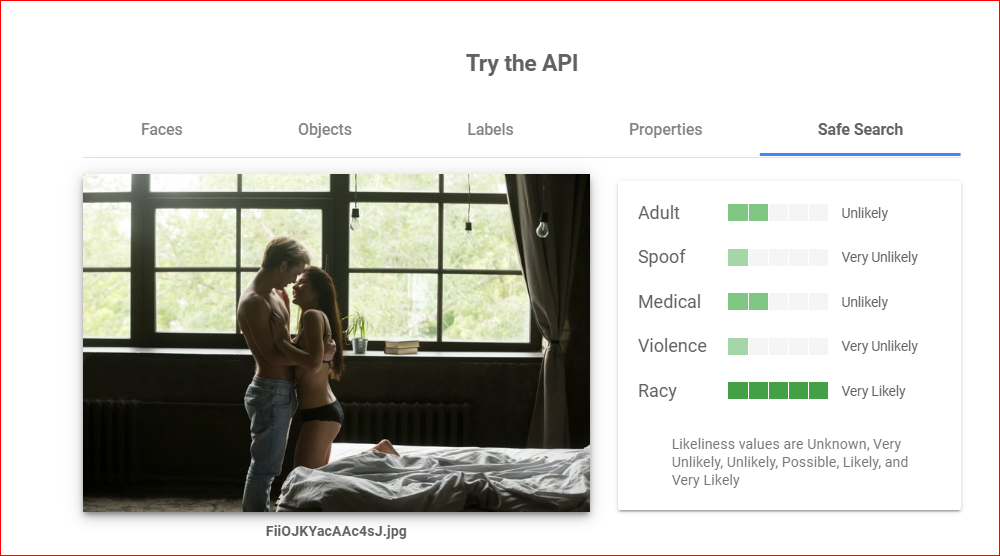
|  |
| --- |
| "safeSearchAnnotation": {  "adult": "UNLIKELY",  "medical": "VERY\_UNLIKELY",  "racy": "VERY\_LIKELY",  "spoof": "UNLIKELY",  "violence": "UNLIKELY" |

**Εικόνα «4»**



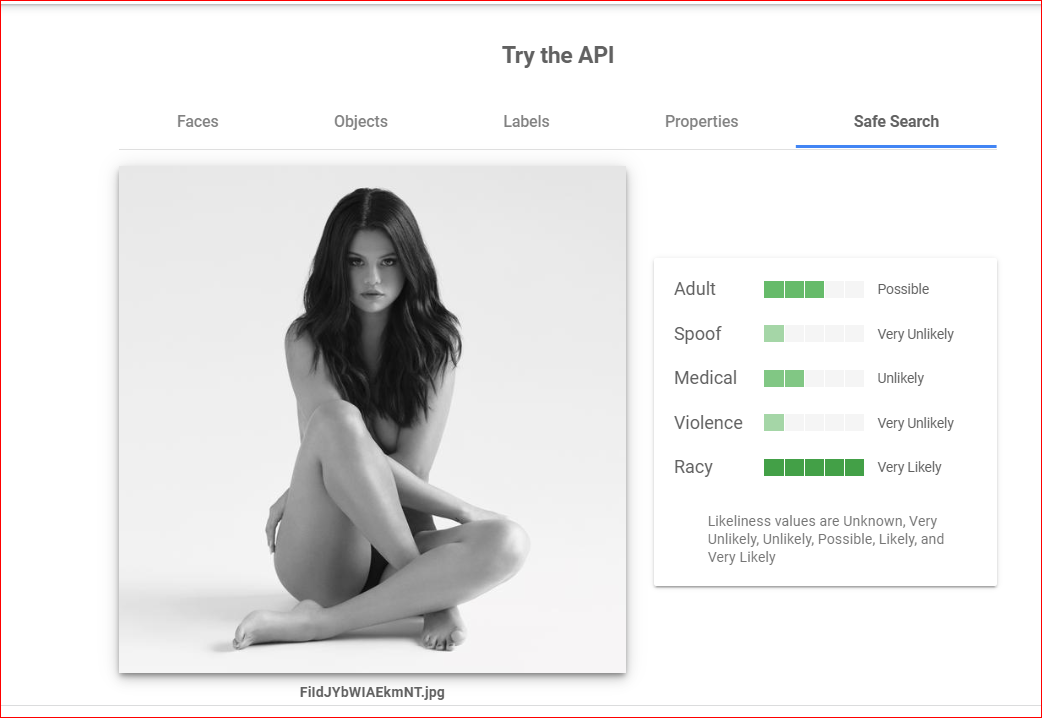
|  |
| --- |
| "safeSearchAnnotation": {  "adult": "VERY\_UNLIKELY",  "medical": "VERY\_UNLIKELY",  "racy": "POSSIBLE",  "spoof": "VERY\_UNLIKELY",  "violence": "UNLIKELY" |

**Εικόνα «5»**



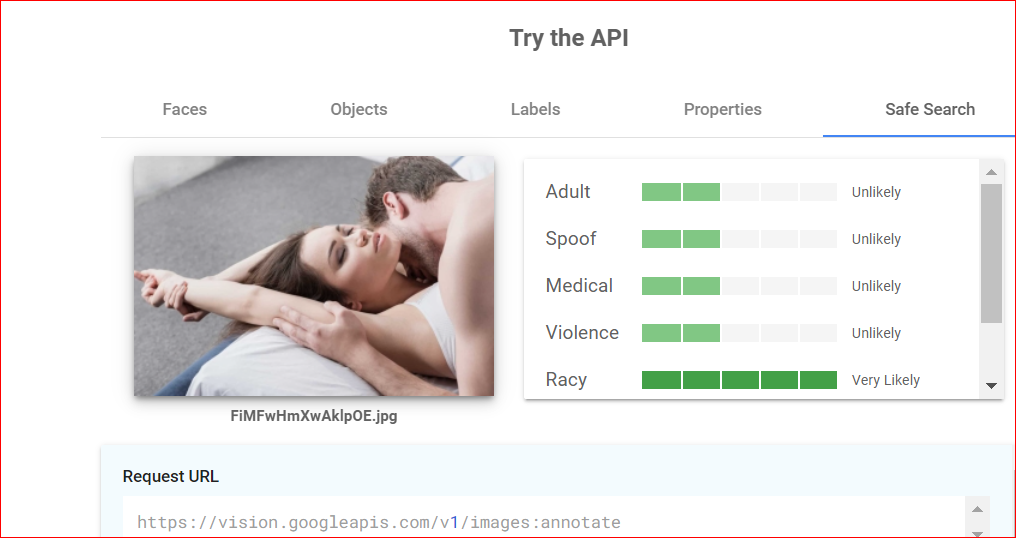
|  |
| --- |
| "safeSearchAnnotation": {  "adult": "UNLIKELY",  "medical": "UNLIKELY",  "racy": "VERY\_LIKELY",  "spoof": "VERY\_UNLIKELY",  "violence": "VERY\_UNLIKELY" |

**Εικόνα «6»**



|  |
| --- |
| "safeSearchAnnotation": {  "adult": "POSSIBLE",  "medical": "UNLIKELY",  "racy": "VERY\_LIKELY",  "spoof": "VERY\_UNLIKELY",  "violence": "VERY\_UNLIKELY" |

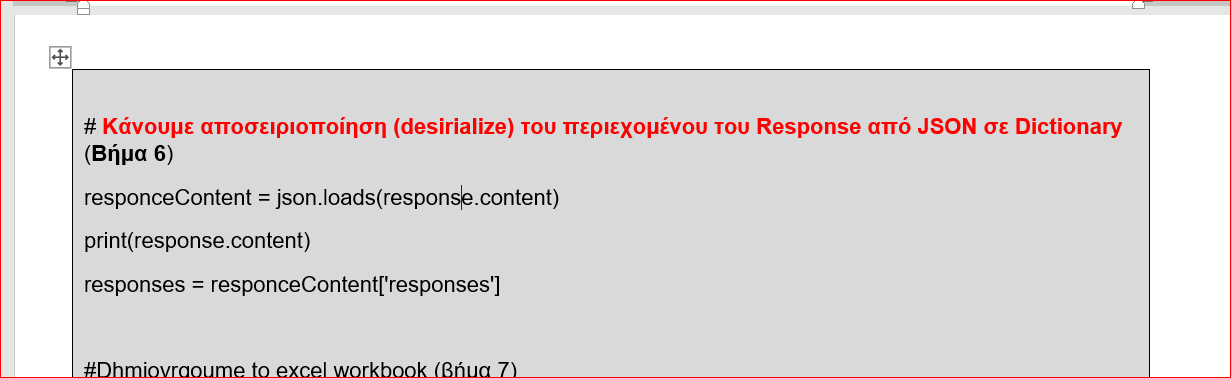
**Εικόνα «7»**



|  |
| --- |
| safeSearchAnnotation": {  "adult": "UNLIKELY",  "medical": "UNLIKELY",  "racy": "VERY\_LIKELY",  "spoof": "UNLIKELY",  "violence": "UNLIKELY" |

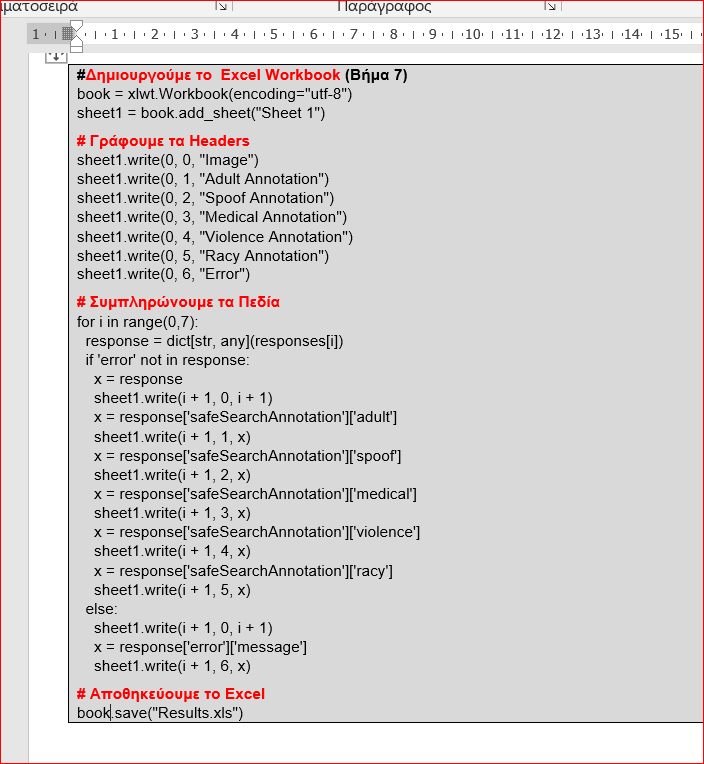
**5o BHMA**

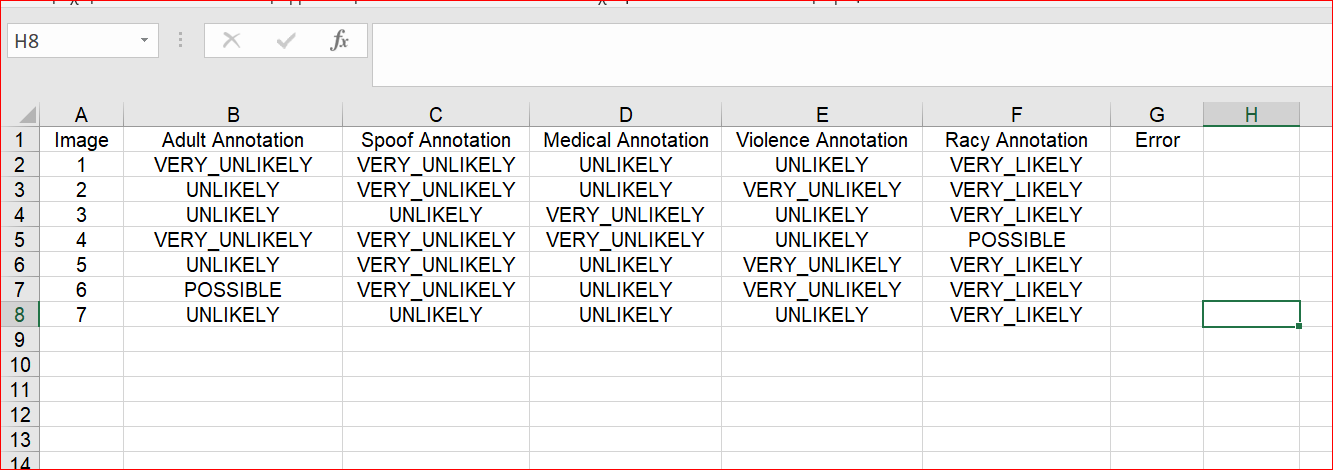
**Επεξεργασία αρχείου απάντησης (response) σε μορφή Json, με σκοπό να μετατραπεί σε πίνακα Εxcel[[3]](#footnote-3)**



**6o BHMA**

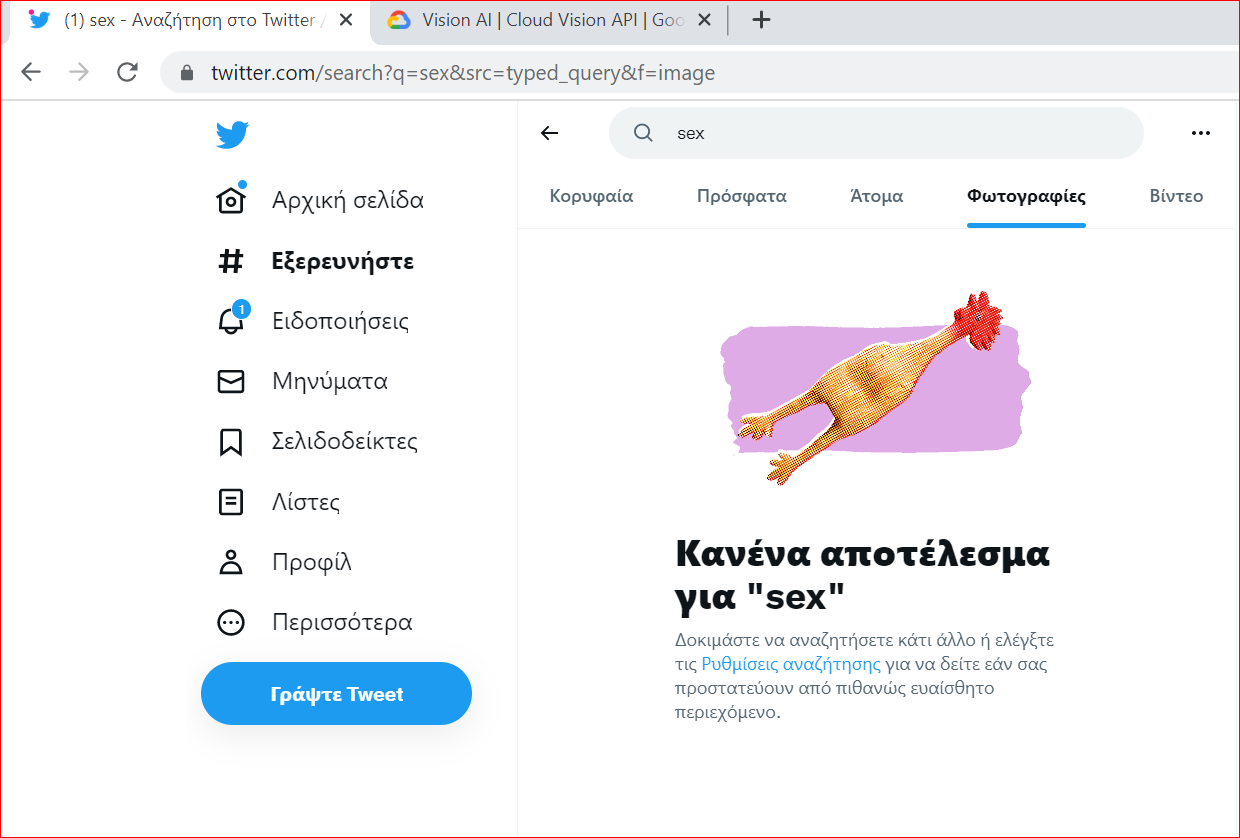
* **Eξαγωγή του επεξεργασμένου αρχείου απάντησης από μορφή Json σε αρχείο excel για περαιτέρω χρήση.**

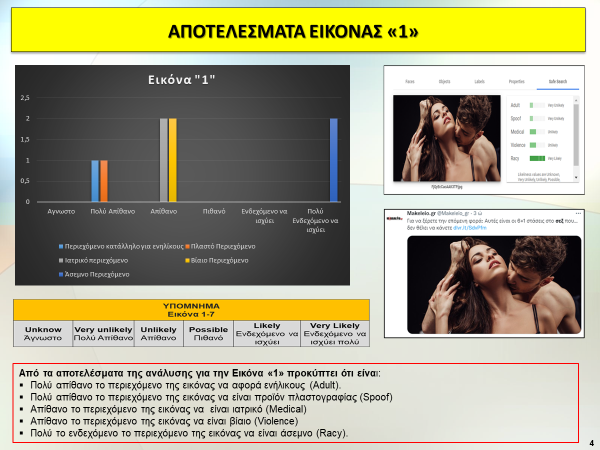


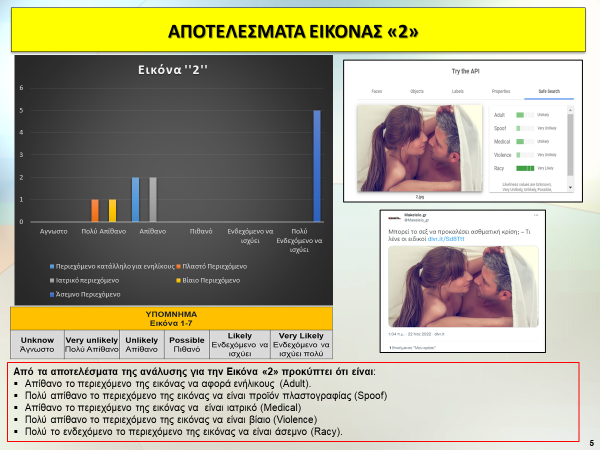


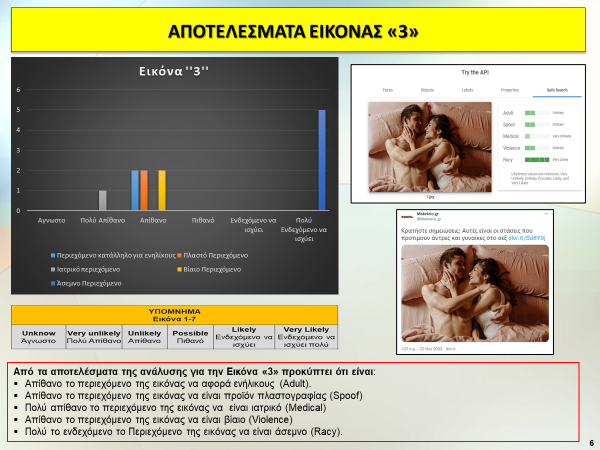
**ΑΝΑΛΥΣΗ - ΣΥΜΠΕΡΑΣΜΑΤΑ**

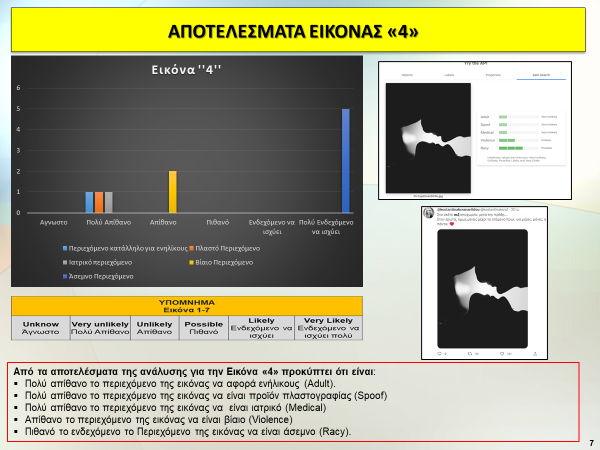
* **Δεν βρέθηκαν αποτελέσματα με αναζήτηση με το ακρωνύμιο εις την αγγλική‘’sex’’**

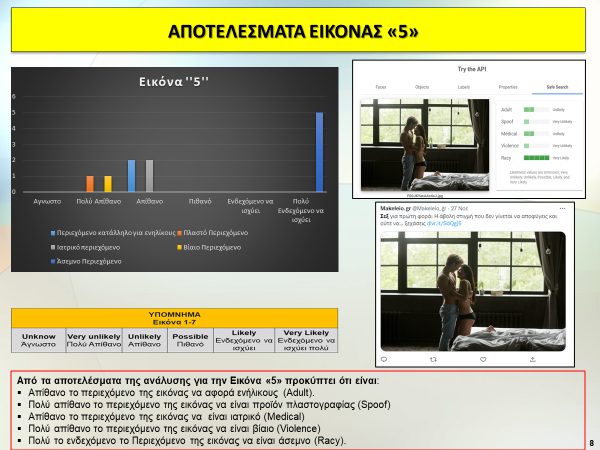


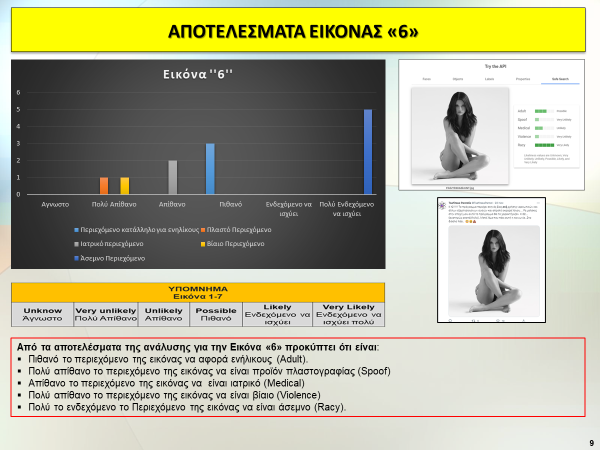


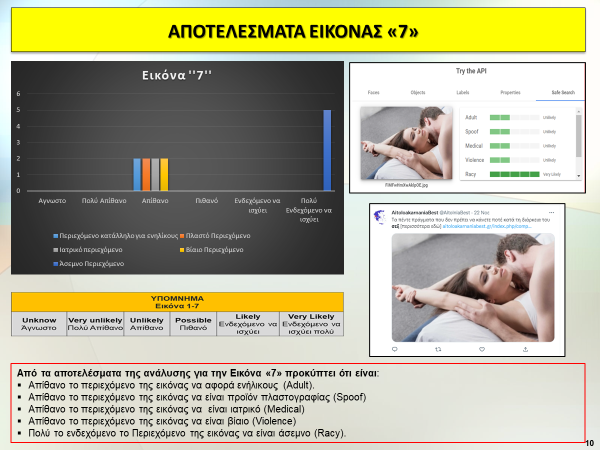




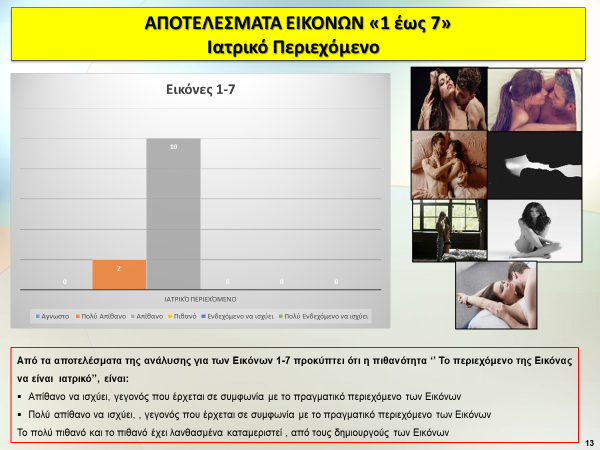


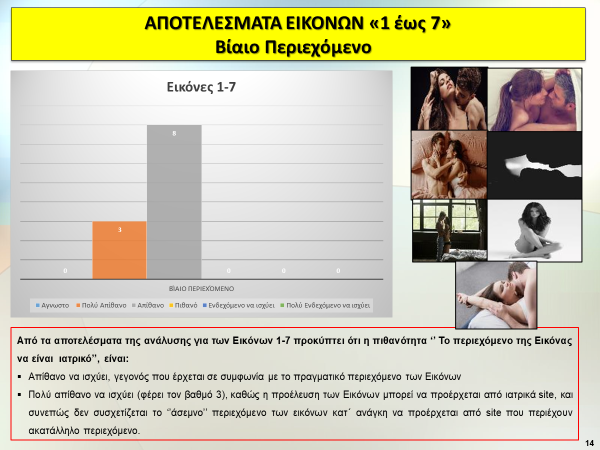


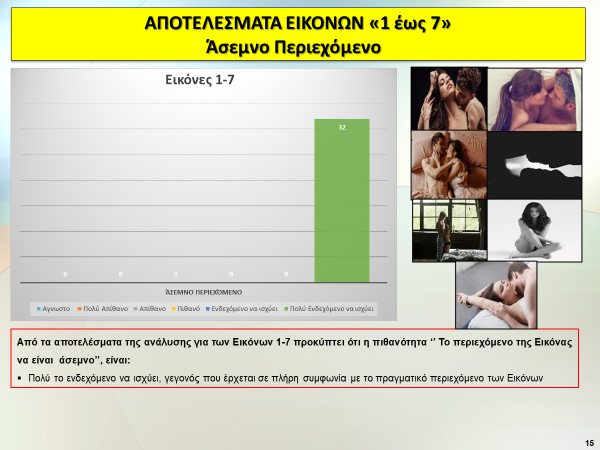














**Κώδικας σε Python**

|  |
| --- |
| import json  import pip.\_vendor.requests  import xlwt  # **Φτιάχνω λίστα με Featureς (Βήμα 3)**  features = [  {  "maxResults": 50,  "type": "LANDMARK\_DETECTION"  },  {  "maxResults": 50,  "type": "FACE\_DETECTION"  },  {  "maxResults": 50,  "type": "OBJECT\_LOCALIZATION"  },  {  "maxResults": 50,  "type": "LOGO\_DETECTION"  },  {  "maxResults": 50,  "type": "LABEL\_DETECTION"  },  {  "maxResults": 50,  "model": "builtin/latest",  "type": "DOCUMENT\_TEXT\_DETECTION"  },  {  "maxResults": 50,  "type": "SAFE\_SEARCH\_DETECTION"  },  {  "maxResults": 50,  "type": "IMAGE\_PROPERTIES"  },  {  "maxResults": 50,  "type": "CROP\_HINTS"  }  ]  #Ftiaxnoume thn prwth eikona  imageSource1 = {  'imageUri': "https://pbs.twimg.com/media/FjQyEcCacAAK3TP?format=jpg&name=medium"  }  image1 = {  'source': imageSource1  }  annotateImageRequest1 = {  "features": features,  'image': image1  }  #Ftiaxnoume thn defterh eikona  imageSource2 = {  'imageUri': "https://pbs.twimg.com/media/FiH4qFsaYAADb9g?format=jpg&name=medium"  }  image2 = {  'source': imageSource2  }  annotateImageRequest2 = {  "features": features,  'image': image2  }  #Ftiaxnoume thn trith eikona  imageSource3 = {  'imageUri': "https://pbs.twimg.com/media/FiIAHZ4aEAI4nRH?format=jpg&name=medium"  }  image3 = {  'source': imageSource3  }  annotateImageRequest3 = {  "features": features,  'image': image3  }  #Ftiaxnoume thn tetarth eikona  imageSource4 = {  'imageUri': "https://pbs.twimg.com/media/Fiv7sp0XoAI2O9A?format=jpg&name=900x900"  }  image4 = {  'source': imageSource4  }  annotateImageRequest4 = {  "features": features,  'image': image4  }  #Ftiaxnoume thn pempth eikona  imageSource5 = {  'imageUri': "https://pbs.twimg.com/media/FiiOJKYacAAc4sJ?format=jpg&name=large"  }  image5 = {  'source': imageSource5  }  annotateImageRequest5 = {  "features": features,  'image': image5  }  #Ftiaxnoume thn ekth eikona  imageSource6 = {  'imageUri': "https://pbs.twimg.com/media/FiIdJYbWIAEkmNT?format=jpg&name=small"  }  image6 = {  'source': imageSource6  }  annotateImageRequest6 = {  "features": features,  'image': image6  }  #Ftiaxnoume thn ebdomh eikona  imageSource7 = {  'imageUri': "https://pbs.twimg.com/media/FiMFwHmXwAklpOE?format=jpg&name=360x360"  }  image7 = {  'source': imageSource7  }  annotateImageRequest7 = {  "features": features,  'image': image7  }  #Ftiaxnoume tto antikeimeno request  requests = [annotateImageRequest1, annotateImageRequest2, annotateImageRequest3, annotateImageRequest4, annotateImageRequest5, annotateImageRequest6, annotateImageRequest7]  requestItem = {  'requests': requests  #Stelnoume to request kai lambanoume to response  response = pip.\_vendor.requests.post("https://vision.googleapis.com/v1/images:annotate?key=AIzaSyB6R\_6-rlKRUo8skDQR8RfsAIHw0POaBTc", json.dumps(requestItem))  #Kanoume desirialize to content toy response apo JSON se Dictionary  responceContent = json.loads(response.content)  print(response.content)  responses = responceContent['responses']  #Dhmioyrgoume to excel workbook  book = xlwt.Workbook(encoding="utf-8")  sheet1 = book.add\_sheet("Sheet 1")  #Grafoume ta Headers  sheet1.write(0, 0, "Image")  sheet1.write(0, 1, "Adult Annotation")  sheet1.write(0, 2, "Spoof Annotation")  sheet1.write(0, 3, "Medical Annotation")  sheet1.write(0, 4, "Violence Annotation")  sheet1.write(0, 5, "Racy Annotation")  sheet1.write(0, 6, "Error")  #Symplhrwnoume ta pedia  for i in range(0,7):  response = dict[str, any](responses[i])  if 'error' not in response:  x = response  sheet1.write(i + 1, 0, i + 1)  x = response['safeSearchAnnotation']['adult']  sheet1.write(i + 1, 1, x)  x = response['safeSearchAnnotation']['spoof']  sheet1.write(i + 1, 2, x)  x = response['safeSearchAnnotation']['medical']  sheet1.write(i + 1, 3, x)  x = response['safeSearchAnnotation']['violence']  sheet1.write(i + 1, 4, x)  x = response['safeSearchAnnotation']['racy']  sheet1.write(i + 1, 5, x)  else:  sheet1.write(i + 1, 0, i + 1)  x = response['error']['message']  sheet1.write(i + 1, 6, x)  #Apothikevoume to Excel  book.save("Results.xls") |

1. [https://cloud.google.com/vision/docs/reference/rest/v1/AnnotateImageRequest#Image](https://cloud.google.com/vision/docs/reference/rest/v1/AnnotateImageRequest%23Image) [↑](#footnote-ref-1)
2. <https://cloud.google.com/vision/docs/reference/rest/v1/images/annotate> [↑](#footnote-ref-2)
3. <https://cloud.google.com/vision/docs/reference/rest/v1/AnnotateImageResponse> [↑](#footnote-ref-3)