Αναφορά Απαλλακτικής Εργασίας στο Μάθημα "Αναγνώριση Προτύπων"

Εργασία του φοιτητή: Μυλωνά Αθανάσιου e21118

Landmark Recognition Project

1. Εισαγωγή

Θέμα

Η εργασία αυτή εστιάζει στην **αναγνώριση μνημείων** μέσα από εικόνες, συνδυάζοντας σύγχρονες τεχνικές μηχανικής μάθησης και υπολογιστικής όρασης. Χρησιμοποιώντας το προεκπαιδευμένο μοντέλο **CLIP** της OpenAI, το οποίο έχει μάθει γενικές αναπαραστάσεις εικόνας-κειμένου από τεράστια σύνολα δεδομένων, πραγματοποιείται fine-tuning με την προσθήκη ενός Linear Classifier. Το αποτέλεσμα είναι ένα σύστημα που μπορεί να αναγνωρίσει με ακρίβεια μνημεία σε νέες εικόνες.

Στόχοι

Οι βασικοί στόχοι της εργασίας περιλαμβάνουν:

- **Ανάπτυξη συστήματος αναγνώρισης μνημείων**:
  - Εφαρμογή του CLIP για εξαγωγή χαρακτηριστικών.
  - Fine-tuning με ένα Linear Classifier για κατηγοριοποίηση εικόνων σε συγκεκριμένα μνημεία.
- **Προετοιμασία και διαχείριση dataset**:
  - Φιλτράρισμα και καθαρισμός των δεδομένων: αφαίρεση κατεστραμμένων εικόνων και κλάσεων με ανεπαρκή δείγματα.
  - Δημιουργία ενός καθαρού, αξιόπιστου dataset για εκπαίδευση και αξιολόγηση.
- **Υλοποίηση φιλικής διεπαφής χρήστη (UI)**:

    - Ανάπτυξη μιας Streamlit web εφαρμογής για ανέβασμα εικόνων ή χρήση κάμερας.
    - Προβολή αποτελεσμάτων με φιλικά ονόματα μνημείων μέσω ενός συστήματος “label mapping”.
- **Αξιολόγηση και βελτιστοποίηση του μοντέλου**:
    - Επίτευξη υψηλής ακρίβειας (Top-1 και Top-5) στο εκπαιδευτικό και δοκιμαστικό σύνολο.
    - Εφαρμογή τεχνικών Early Stopping και Learning Rate Scheduling για αποφυγή overfitting.

Περιορισμοί

Κατά την υλοποίηση αντιμετωπίστηκαν συγκεκριμένοι περιορισμοί:

- **Υποσύνολο Δεδομένων**:
    - Χρησιμοποιήθηκε μειωμένο υποσύνολο του Google Landmarks Dataset v2 για λόγους αποθήκευσης και υπολογιστικής ισχύος, περιορίζοντας έτσι τη γενικότητα του μοντέλου.
- **Απλοποίηση Dataset**:
    - Εξαιρέθηκαν κλάσεις με λιγότερα από 10 δείγματα για να διασφαλιστεί η επάρκεια δεδομένων.
- **Υποδομή Υλοποίησης**:
    - Η εκπαίδευση πραγματοποιήθηκε στο Google Colab, με περιορισμούς χρόνου εκτέλεσης, GPU (π.χ. NVIDIA T4) και αποθηκευτικού χώρου.
- **Απαιτήσεις Χρήστη**:
    - Η χρήση της εφαρμογής προϋποθέτει βασικές γνώσεις χειρισμού αρχείων και σύνδεση στο διαδίκτυο για τη χρήση του Streamlit.

2. Περιγραφή Δεδομένων

Πηγή Δεδομένων: Το project βασίστηκε στο **Google Landmarks Dataset v2**, ένα από τα μεγαλύτερα δημόσια διαθέσιμα datasets για την αναγνώριση μνημείων, που περιέχει περίπου 5 εκατομμύρια εικόνες σε ~81.313 κατηγορίες (landmark_id). Το σύνολο δεδομένων καλύπτει χιλιάδες μοναδικά μνημεία και τοποθεσίες σε όλο τον κόσμο.

Υποσύνολο Δεδομένων: Για λόγους αποδοτικότητας, χρησιμοποιήθηκε ένα μικρότερο υποσύνολο:

- **Train set:** 20 αρχεία TAR (~1GB/αρχείο).
- **Index set:** 3 αρχεία (~850MB/αρχείο).
- **Test set:** 2 αρχεία (~500MB/αρχείο).

Συνολικά, χρησιμοποιήθηκαν περίπου 25GB δεδομένων.

Επεξεργασία και Φιλτράρισμα: Η προετοιμασία του dataset περιλάμβανε τα εξής βήματα:

- **Έλεγχος για ελλιπή/κατεστραμμένα δεδομένα:**
  - Αφαιρέθηκαν εικόνες που έλειπαν ή ήταν κατεστραμμένες.
  - Από τα 165.320 δείγματα που εξετάστηκαν, 4.799 εικόνες διαπιστώθηκαν ως ελλείπουσες, ενώ οι υπόλοιπες ήταν έγκυρες.
- **Αποκλεισμός κατηγοριών με λίγα δείγματα:**
  - Εξαιρέθηκαν κατηγορίες (landmark_id) με λιγότερες από 10 εικόνες.
  - Από τις αρχικές 75.102 κατηγορίες διατηρήθηκαν 1.434 με επαρκή δεδομένα.
- **Τελικό Dataset:**
  - **Εικόνες:** 23.865.
  - **Κατηγορίες:** 1.434.
  - Τα δεδομένα χωρίστηκαν σε training και validation set σε αναλογία 80% - 20%.

Τεχνολογίες και Μεθοδολογίες:

- **Pandas:** Χρήση για φόρτωση, φιλτράρισμα και ανάλυση metadata.

```
valid_landmarks =
filtered_metadata['landmark_id'].value_counts()
valid_landmarks = valid_landmarks[valid_landmarks >=
10].index
final_filtered_metadata =
filtered_metadata[filtered_metadata['landmark_id'].isin(valid
_landmarks)]
```

- **glob και os:** Εντοπισμός εικόνων σε φακέλους και έλεγχος ύπαρξης αρχείων.

```
missing_images = [img_id for img_id in metadata['id'] if not
os.path.exists(f"path/to/images/{img_id}.jpg")]
```

- **PIL και tqdm:** Έλεγχος για κατεστραμμένες εικόνες με progress bars για διαχείριση μεγάλου όγκου δεδομένων.
- **Αποθήκευση:** Δημιουργήθηκαν τα final_filtered_train.csv και label_mapping.json για χρήση στην εκπαίδευση και το inference.

Σκοπός της Επεξεργασίας: Η προσεκτική επεξεργασία και το φιλτράρισμα ήταν ουσιώδη για να:

- Εξασφαλιστεί ότι το μοντέλο εκπαιδεύεται σε υψηλής ποιότητας δεδομένα.
- Μειωθεί ο θόρυβος στο training, οδηγώντας σε καλύτερη απόδοση και γενίκευση.

3. Μοντέλο

CLIP Backbone

Τι είναι το CLIP: Το **CLIP** (Contrastive Language–Image Pretraining) της OpenAI είναι ένα προεκπαιδευμένο μοντέλο που συνδυάζει τεχνικές μηχανικής μάθησης και φυσικής γλώσσας για να κατανοήσει και να συσχετίσει εικόνες με κείμενα. Εκπαιδεύτηκε σε 400 εκατομμύρια ζεύγη εικόνας-κειμένου, αποκτώντας ικανότητα δημιουργίας ενσωματώσεων (embeddings) που αντικατοπτρίζουν οπτικές και γλωσσικές πληροφορίες. Αυτό το καθιστά εξαιρετικά ευέλικτο για διάφορες εργασίες όπως η αναγνώριση αντικειμένων και το zero-shot learning.

Γιατί επιλέχθηκε:

- **Ισχυρή προεκπαίδευση:** Το CLIP έχει μάθει πλούσια χαρακτηριστικά από ένα τεράστιο σύνολο δεδομένων, μειώνοντας τις απαιτήσεις σε δεδομένα για το fine-tuning.
- **Γενικότητα και ευελιξία:** Η ικανότητά του να εξάγει γενικευμένες αναπαραστάσεις καθιστά δυνατή την προσαρμογή σε διαφορετικές εργασίες με λιγότερη επιπλέον εκπαίδευση.
- **Εξαγωγή υψηλής ποιότητας χαρακτηριστικών:** Τα embeddings που παράγει είναι κατάλληλα για προβλήματα ταξινόμησης, προσφέροντας πλούσιες και διακριτικές αναπαραστάσεις.

Χαρακτηριστικά CLIP Backbone:

- **Αρχιτεκτονική:** Χρησιμοποιήθηκε η έκδοση **ViT-B/32** (Vision Transformer), η οποία χωρίζει κάθε εικόνα σε patches 32x32 pixels και τα επεξεργάζεται μέσω 12 transformer layers. Κάθε transformer layer αποτελείται από:
    - **Multi-Head Self-Attention Layers:** Επιτρέπουν στο μοντέλο να εστιάσει σε διαφορετικά μέρη της εικόνας ταυτόχρονα.
    - **Feedforward Layers:** Επεξεργάζονται τις εξόδους του self-attention, εισάγοντας μη γραμμικότητα και βελτιώνοντας την εκμάθηση σύνθετων μοτίβων.
- Αυτή η πολυεπίπεδη δομή επιτρέπει το CLIP να εξάγει τόσο βασικά όσο και πολύπλοκα οπτικά χαρακτηριστικά από τις εικόνες.
- **Embeddings 512 Διαστάσεων:**

Το CLIP εξάγει για κάθε εικόνα ένα **512-διάστατο διάνυσμα** – ένα υψηλής διάστασης embedding που συμπιέζει πλούσιες οπτικές πληροφορίες της εικόνας. Αυτές οι 512 διαστάσεις αποτελούν την τελική έξοδο του CLIP backbone και χρησιμεύουν ως είσοδος για τον επόμενο Linear Classifier.

Γιατί 512 διαστάσεις;
Η επιλογή του μεγέθους των 512 διαστάσεων αποτελεί ισορροπία μεταξύ ικανότητας αναπαράστασης και υπολογιστικής αποδοτικότητας. Με αυτόν τον αριθμό, το μοντέλο έχει αρκετό «χώρο» να συλλάβει πολύπλοκα μοτίβα και λεπτομέρειες, χωρίς να αυξάνει υπερβολικά τον υπολογιστικό φόρτο. Αυτός ο αριθμός δίνει στο μοντέλο την επαρκή χωρητικότητα να αναγνωρίζει διαφορές και ομοιότητες μεταξύ των μνημείων, βελτιώνοντας την ακρίβεια κατά την ταξινόμηση.

Τι περιέχει το διάνυσμα;

- Κάθε μία από τις 512 διαστάσεις δεν αντιστοιχεί σε ένα συγκεκριμένο, απλό οπτικό χαρακτηριστικό (όπως χρώμα ή σχήμα). Αντίθετα, κάθε διάσταση συμβάλλει σε μια σύνθετη, υψηλής διάστασης αναπαράσταση που συνδυάζει πληθώρα πληροφοριών για φόρμες, υφές, χρώματα και πιο αφηρημένα μοτίβα.
- Οι διαστάσεις αυτές αποτελούν το αποτέλεσμα της μάθησης που έχει πραγματοποιήσει το CLIP σε δισεκατομμύρια εικόνες-κειμένων, καταγράφοντας πολύπλοκα μοτίβα και σχέσεις μέσα στα δεδομένα.

Σημασία στον Linear Classifier:
Αυτά τα 512-διάστατα embeddings χρησιμοποιούνται ως είσοδος για τον Linear Classifier. Ο classifier μαθαίνει να μετατρέπει αυτά τα πλούσια, συμπιεσμένα

διανύσματα σε συγκεκριμένες κατηγορίες μνημείων. Η υψηλή διάσταση του embedding εξασφαλίζει ότι το μοντέλο διατηρεί λεπτομερείς πληροφορίες που είναι απαραίτητες για τη διακριτή αναγνώριση χιλιάδων διαφορετικών μνημείων, ενώ η επιλογή του αριθμού των διαστάσεων εξασφαλίζει αποτελεσματικότητα τόσο στην εκπαίδευση όσο και στο inference.

Linear Classifier

Προσθήκη Linear Classifier στο CLIP:

- Το CLIP μόνο του δεν ταξινομεί εικόνες σε συγκεκριμένες κατηγορίες, επομένως προστέθηκε ένας **Linear Classifier** στο τέλος του CLIP.
- **Αρχιτεκτονική:**
  - **Είσοδος:** Τα 512-διάστατα embeddings από το CLIP.
  - **Έξοδος:** 1434 κατηγορίες, οι οποίες αντιστοιχούν στα καθαρισμένα landmarks μετά το φιλτράρισμα.
  - **Dropout:** Χρησιμοποιείται με ρυθμό 0.1 για αποφυγή overfitting.

Κλάση CLIPWithClassifier: Ο κώδικας της κλάσης ενσωματώνει το CLIP με έναν Linear Classifier, προσφέροντας επιλογές όπως το freeze των βαρών του CLIP:

```
import torch.nn as nn

class CLIPWithClassifier(nn.Module):
     def __init__(self, clip_model, num_classes,
dropout_rate=0.1, freeze_clip=False):
     super(CLIPWithClassifier, self).__init__()
     self.clip_model = clip_model
     self.dropout = nn.Dropout(dropout_rate)
     self.classifier =
nn.Linear(self.clip_model.visual.output_dim, num_classes)

     if freeze_clip:
          for param in self.clip_model.parameters():
               param.requires_grad = False
```

```
    def forward(self, images):
    image_features = self.clip_model.encode_image(images)
    x = self.dropout(image_features)
    logits = self.classifier(x)
    return logits
```

Fine-tuning του Μοντέλου

Προσέγγιση Fine-tuning:

- **Freeze/Unfreeze Layers:** Στην αρχή, τα βάρη του CLIP encoder μπορούν να παγώσουν για να διατηρηθούν οι γενικευμένες αναπαραστάσεις, ενώ εκπαιδεύεται ο Linear Classifier. Μετά, επιλεγμένα layers του CLIP μπορούν να ξεπαγώσουν για καλύτερη προσαρμογή στις νέες κατηγορίες.
- **Precision (Float32):** Η εκπαίδευση έγινε σε float32 για να διασφαλιστεί ακρίβεια και σταθερότητα.

Τεχνικές Εκπαίδευσης:

- **Gradient Clipping:** Χρησιμοποιήθηκε για σταθερότητα κατά το backpropagation, περιορίζοντας τα gradients με max norm 1.0.
- **Early Stopping & LR Scheduler:** Παρακολουθούν την απόδοση στο validation set, μειώνοντας το learning rate όταν χρειάζεται και σταματώντας την εκπαίδευση όταν οι βελτιώσεις γίνονται αμελητέες.

Διαδικασία Εκπαίδευσης:

1. **Φόρτωση του προεκπαιδευμένου CLIP:**

```
import clip
clip_model, _ = clip.load("ViT-B/32", device=device,
jit=False)
clip_model = clip_model.float()
```

2. **Ενσωμάτωση του Linear Classifier:**

```
num_classes = 1434
model = CLIPWithClassifier(clip_model=clip_model,
num_classes=num_classes, dropout_rate=0.1,
```

```
freeze_clip=False).to(device)
```

3. **Φόρτωση βαρών από το checkpoint:**

```
checkpoint = torch.load("models/BestClipModel.pth",
map_location=device)
model.load_state_dict(checkpoint["model_state_dict"])
model.eval()
```

4. **Εκπαίδευση:** Ορίζονται optimizer, loss function, scheduler και τρέχει το training loop με τις παραπάνω τεχνικές για σταθερή και αποδοτική εκπαίδευση.

Αποτελεσματικότητα

Με αυτήν την προσέγγιση, ο συνδυασμός του ισχυρού CLIP backbone με τον απλό Linear Classifier καταφέρνει:

- **Υψηλή ακρίβεια:** Top-1 ακρίβεια άνω του 98% και Top-5 άνω του 99% στο validation set.
- **Ταχύτητα Inference:** Η χρήση του CLIP ως feature extractor επιτρέπει γρήγορη πρόβλεψη νέων εικόνων, καθιστώντας το μοντέλο κατάλληλο για πραγματικούς χρόνους.
- **Γενίκευση:** Οι πολυάριθμες διαστάσεις των embeddings παρέχουν ισχυρή ικανότητα γενίκευσης σε νέες, ποικίλες εικόνες, διατηρώντας υψηλή απόδοση σε διαφορετικές συνθήκες.

4. Εκπαίδευση

Μεθοδολογία

Loss Function:

Η CrossEntropyLoss χρησιμοποιήθηκε για την πολυκατηγορική ταξινόμηση, συγκρίνοντας τις προβλεπόμενες πιθανότητες του μοντέλου με τις πραγματικές κατηγορίες. Αυτή η συνάρτηση μετράει τη διαφορά μεταξύ των διανεύσματος εξόδου του μοντέλου (μετά το softmax) και της one-hot κωδικοποίησης της σωστής κατηγορίας.

Optimizer:
Ο **Adam Optimizer** επιλέχθηκε λόγω της ικανότητάς του να προσαρμόζει δυναμικά το learning rate για κάθε παράμετρο, συνδυάζοντας τα πλεονεκτήματα των μεθόδων όπως το RMSProp και το Momentum. Αυτό βοηθάει στην ταχύτερη σύγκλιση και την αποτελεσματική διαχείριση των gradients σε μεγάλα και πολύπλοκα δίκτυα.

Learning Rate Scheduler:
Ο **ReduceLROnPlateau** μειώνει το learning rate όταν το validation loss δεν παρουσιάζει βελτίωση για κάποιο χρονικό διάστημα, επιτρέποντας στο μοντέλο να κάνει πιο λεπτομερή προσαρμογή των παραμέτρων του όσο σταθεροποιείται η απόδοση.

Early Stopping:
Ο μηχανισμός **Early Stopping** τερματίζει την εκπαίδευση αν το validation loss δεν βελτιώνεται για πέντε συνεχόμενες εποχές, μειώνοντας έτσι τον κίνδυνο υπερπροσαρμογής (overfitting) και εξοικονομώντας υπολογιστικούς πόρους.

Διαδικασία Εκπαίδευσης

- **Training/Validation Split:**
  Το dataset χωρίστηκε σε 80% για εκπαίδευση και 20% για επαλήθευση, χρησιμοποιώντας διαστρωματωμένη δειγματοληψία ώστε να διατηρηθεί η αναλογία των κλάσεων και στα δύο σύνολα.
- **Batch Size:**
  Το μέγεθος του batch ορίστηκε στα 32 για αποτελεσματική χρήση μνήμης και σταθερή ενημέρωση των βαρών.
- **Αριθμός Εποχών:**
  Η εκπαίδευση πραγματοποιήθηκε για συνολικά 26 εποχές με εφαρμογή early stopping και learning rate scheduling. Αν και σε σημείο παρατηρήθηκαν πολύ υψηλές μετρικές, η διαδικασία συνεχίστηκε μέχρι το τέλος λόγω των βελτιώσεων στο validation loss.

Νευρωνικό Δίκτυο και Layers

Δομή Νευρωνικού Δικτύου:

- Το μοντέλο αποτελείται από δύο βασικά μέρη:
    1. **CLIP Backbone (ViT-B/32):** Ένα βαθύ νευρωνικό δίκτυο βασισμένο σε Vision Transformer (ViT) που περιέχει 12 transformer layers. Κάθε transformer layer αποτελείται από:
        - **Multi-Head Self-Attention Layers:** Επιτρέπουν στο μοντέλο να εστιάσει σε διάφορα μέρη της εικόνας ταυτόχρονα, καταγράφοντας σχέσεις μεταξύ τους.
        - **Feedforward Layers:** Επεξεργάζονται τις εξόδους του self-attention και εισάγουν μη γραμμικότητα, ενισχύοντας την ικανότητα μάθησης πολύπλοκων χαρακτηριστικών.
    2. **Linear Classifier:** Ένας απλός πλήρως συνδεδεμένος layer (με dropout), ο οποίος λαμβάνει τα 512-διάστατα embeddings από το CLIP και τα ταξινομεί σε μία από τις 1434 κατηγορίες μνημείων.

Γιατί τόσα layers;

- **Ιεραρχική Μάθηση:** Κάθε επιπλέον layer επιτρέπει στο δίκτυο να εξάγει πιο αφηρημένα και σύνθετα χαρακτηριστικά. Τα πρώτα layers συλλαμβάνουν βασικές οπτικές πληροφορίες (όρια, χρώματα), ενώ τα πιο βαθιά layers μαθαίνουν πιο σύνθετες έννοιες όπως μέρη αντικειμένων και σχέσεις μεταξύ τους.
- **Αυξημένη Ικανότητα Εκμάθησης:** Περισσότερα layers σημαίνουν μεγαλύτερη χωρητικότητα μοντέλου, επιτρέποντας την αναγνώριση πιο πολύπλοκων μοτίβων και λεπτομερειών που είναι απαραίτητα για την ταξινόμηση σε πολλές κατηγορίες, όπως 1434 διαφορετικά μνημεία.
- **Embeddings 512 Διαστάσεων:** Το τελικό διάνυσμα 512 διαστάσεων εξάγεται από το τελευταίο layer του CLIP, το οποίο έχει συλλέξει πληροφορίες μέσω όλων των προηγούμενων layers. Οι 512 διαστάσεις αποτελούν μια συμπιεσμένη αναπαράσταση της εικόνας, όπου κάθε διάσταση συνεισφέρει στη συνολική κατανόηση του περιεχομένου χωρίς να αντιστοιχεί άμεσα σε μια συγκεκριμένη οπτική ιδιότητα.

Fine-tuning του Μοντέλου

Προσέγγιση Fine-tuning:

- **Freeze/Unfreeze Layers:** Στην αρχή του fine-tuning μπορεί να παγώσει ολόκληρο το CLIP, εκπαιδεύοντας μόνο τον Linear Classifier. Μετά από

μερικές εποχές, επιλέγεται να «ξεπαγώσει» σταδιακά τα τελευταία transformer layers για λεπτομερέστερη προσαρμογή στις συγκεκριμένες κλάσεις του dataset.
- **Precision (Float32):** Η εκπαίδευση πραγματοποιήθηκε σε float32 για να διασφαλιστεί η αριθμητική ακρίβεια και να αποφευχθούν προβλήματα που μπορεί να προκύψουν από τη χρήση float16.

Τεχνικές Εκπαίδευσης:

- **Gradient Clipping:** Εφαρμογή με μέγιστη τιμή 1.0 για σταθερότητα των gradients.
- **Early Stopping:** Διακοπή εκπαίδευσης αν δεν υπάρχει σημαντική βελτίωση στο validation loss για 5 συνεχόμενες εποχές.
- **LR Scheduler:** Χρήση του ReduceLROnPlateau για μείωση του learning rate όταν το validation loss σταθεροποιείται.

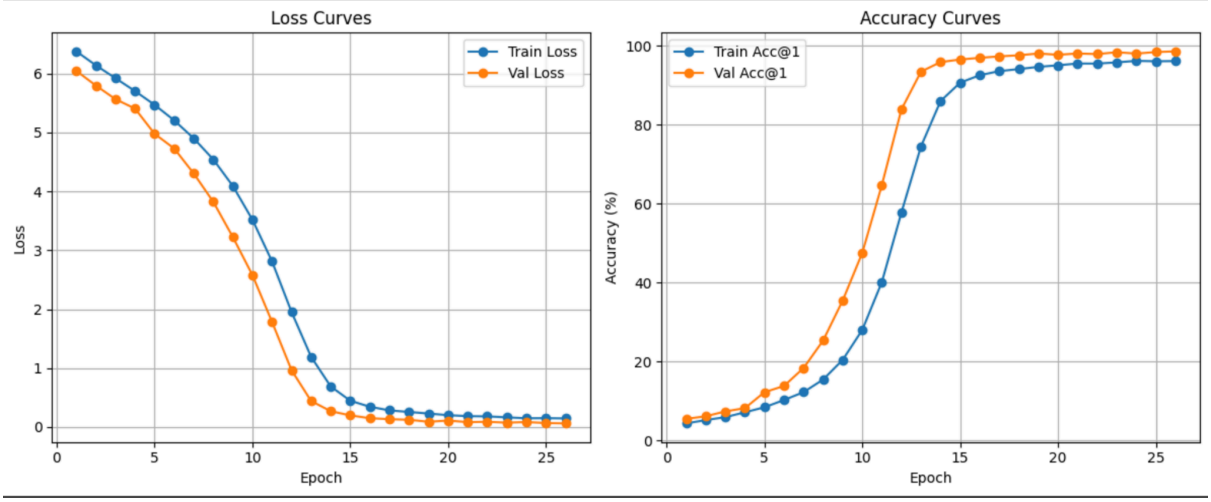
Διαδικασία:

1. **Φόρτωση και Προσαρμογή του CLIP:** Χρησιμοποιήθηκε η έκδοση ViT-B/32 που φορτώθηκε σε float32.
2. **Προσθήκη του Linear Classifier:** Ο classifier προσαρμόστηκε για 1434 κατηγορίες και ενσωματώθηκε πάνω από το CLIP.
3. **Εκπαίδευση:** Το μοντέλο εκπαιδεύτηκε χρησιμοποιώντας τα παραπάνω τεχνικά, παρακολουθώντας training και validation metrics σε κάθε epoch.

Αποτελέσματα Εκπαίδευσης

Learning Curves & Accuracy:

- **Loss Curves:** Το training loss μείωσε σταθερά, με το validation loss να σταθεροποιείται μετά τις 15 εποχές.
- **Accuracy Curves:** Η ακρίβεια στο validation set αυξήθηκε δραστικά τις πρώτες εποχές, φτάνοντας 98.58% Top-1 ακρίβεια και 99.85% Top-5 ακρίβεια στα τελευταία epochs.



Συζήτηση Αποτελεσμάτων:

- Το μοντέλο σταθεροποιήθηκε γύρω στην 16η εποχή, με τη χρήση Early Stopping και LR Scheduler να βοηθούν στην αποφυγή υπερπροσαρμογής.
- Η υψηλή ακρίβεια αποδεικνύει ότι ο συνδυασμός του CLIP με τον Linear Classifier είναι αποτελεσματικός για την αναγνώριση μνημείων.

Παράλληλη Σχόλια για Layers:

- Ο σχεδιασμός με πολλαπλά layers στον transformer (όπως στη ViT-B/32) επιτρέπει την εξαγωγή βαθύτατων χαρακτηριστικών που βοηθούν σημαντικά στην κατηγοριοποίηση, ειδικά όταν υπάρχουν πολλές κατηγορίες.
- Οι διάφορες βαθμίδες των layers δουλεύουν συλλογικά, καταγράφοντας τόσο βασικά όσο και πολύπλοκα μοτίβα, και τελικά αποδίδουν το 512-διάστατο embedding που ο classifier χρησιμοποιεί για να ταξινομήσει τις εικόνες.

5. Αξιολόγηση

Metrics

Το μοντέλο αξιολογήθηκε με δύο βασικά μετρικά:

- **Top-1 Accuracy:** Το ποσοστό των περιπτώσεων όπου η κορυφαία πρόβλεψη του μοντέλου ήταν σωστή. Το μοντέλο πέτυχε τελικό Top-1 Accuracy 98.58% στο validation set, υποδεικνύοντας ότι στις περισσότερες περιπτώσεις η πρώτη πρόβλεψη του ήταν η σωστή κατηγορία.

- **Top-5 Accuracy:** Το ποσοστό των περιπτώσεων όπου η σωστή κλάση βρισκόταν μεταξύ των πέντε κορυφαίων προβλέψεων. Το μοντέλο πέτυχε Top-5 Accuracy 99.85% στο validation set, επιβεβαιώνοντας ότι ακόμα και όταν η κορυφαία πρόβλεψη ήταν λανθασμένη, η σωστή κατηγορία συμπεριλαμβανόταν συνήθως στις πέντε πρώτες.

Αυτά τα υψηλά νούμερα επιβεβαιώνουν την εξαιρετική ικανότητα του μοντέλου να αναγνωρίζει μνημεία.

Overfitting/Underfitting

Η ανάλυση των learning curves υποδεικνύει καλή ισορροπία:

- **Training Loss:** Μειώνεται σταθερά και τελικά σταθεροποιείται.
- **Validation Loss:** Ακολουθεί παρόμοια πορεία, χωρίς να αρχίσει να αυξάνεται, πράγμα που δείχνει ότι το μοντέλο δεν υπέστη σημαντικό overfitting.

Η χρήση τεχνικών όπως **Early Stopping** και **Learning Rate Scheduler** απέφυγε υπερπροσαρμογή του μοντέλου, τερματίζοντας την εκπαίδευση όταν οι βελτιώσεις σταματούσαν να είναι σημαντικές.

Inference

Στο στάδιο του inference, το μοντέλο εφαρμόστηκε σε πραγματικές εικόνες για αξιολόγηση:

- **Είσοδος:** Μία εικόνα μνημείου από το test set.
- **Προβλέψεις:**
  - **Top-1:** Η πιο πιθανή κλάση (π.χ. class 159) με πιθανότητα 39.57%.
  - **Top-2 έως Top-5:** Άλλες πιθανές κλάσεις με πιθανότητες μεταξύ 4.33% και 9.67%.

Παρατηρήσεις:

- Η κορυφαία πρόβλεψη έχει υψηλή πιθανότητα σε σχέση με τις επόμενες, δίνοντας στο χρήστη την πιο σίγουρη απάντηση.
- Οι επόμενες προβλέψεις παρέχουν λογικές εναλλακτικές, ενισχύοντας την αξιοπιστία σε περιπτώσεις αβεβαιότητας.

Παρά την υψηλή ακρίβεια στο validation, παρατηρείται πως οι πιθανότητες Top-1 δεν είναι πάντα υπερβολικά υψηλές, γεγονός που υποδηλώνει ότι το μοντέλο

διατηρεί κάποια σιγουριά στην πρόβλεψή του αλλά λαμβάνει υπόψη του και άλλες πιθανές κλάσεις στις κορυφαίες επιλογές του.

Overfitting/Underfitting – Ανάλυση

- **Ισορροπία Loss:** Οι καμπύλες training και validation loss παρουσιάζουν σταθερή μείωση χωρίς μεγάλη απόκλιση μεταξύ τους, υποδεικνύοντας ότι το μοντέλο έχει μάθει να γενικεύει καλά στα νέα δεδομένα.
- **Χρήση Early Stopping και LR Scheduler:** Συμβάλλει στην σταθεροποίηση της εκπαίδευσης και στην αποφυγή υπερπροσαρμογής, καθώς παρατηρείται σταθεροποίηση του validation loss μετά από συγκεκριμένο αριθμό εποχών.

Σχόλια για τα Αποτελέσματα Εκπαίδευσης

Παρόλο που δεν μπορούμε να δούμε το γράφημα στο έγγραφο, τα **logs** εκπαίδευσης δείχνουν ότι:

- Οι μετρικές βελτιώθηκαν σταθερά, με το validation loss να μειώνεται κατακόρυφα τις πρώτες 10–15 εποχές.
- Μετά την 16η εποχή, οι μετρικές σταθεροποιήθηκαν, γεγονός που υποδηλώνει ότι το μοντέλο είχε φτάσει σε όριο βελτίωσης μέσα στο παρόν dataset.

Αυτά τα αποτελέσματα επιβεβαιώνουν την αποτελεσματικότητα της μεθοδολογίας εκπαίδευσης και την ικανότητα του μοντέλου να γενικεύει καλά, χωρίς να υπέστη overfitting.

6. Πλεονεκτήματα και Περιορισμοί

Πλεονεκτήματα:

- **Χρήση CLIP:**
  Η αξιοποίηση του προεκπαιδευμένου μοντέλου CLIP εξασφαλίζει ισχυρή αρχική γνώση από τεράστια σύνολα δεδομένων, προσφέροντας υψηλή ακρίβεια στην αναγνώριση. Ο συνδυασμός εικόνας και κειμένου μέσω των text-image embeddings καθιστά το μοντέλο ευέλικτο και ικανό να αντιμετωπίζει ποικίλα σενάρια αναγνώρισης.
- **Εύκολη Προσαρμογή:**
  Η ενσωμάτωση ενός απλού Linear Classifier πάνω από το CLIP επιτρέπει γρήγορο fine-tuning. Νέα δεδομένα και κατηγορίες μπορούν να

ενσωματωθούν εύκολα με επανεκπαίδευση μόνο του classifier, χωρίς την ανάγκη για εκ νέου εκπαίδευση ολόκληρου του μοντέλου. Αυτό μειώνει το χρόνο και τους υπολογιστικούς πόρους που απαιτούνται για προσαρμογή σε νέες απαιτήσεις.

Περιορισμοί:

- **Μέγεθος Dataset:**
  Λόγω περιορισμών πόρων χρησιμοποιήθηκε υποσύνολο των πλήρων δεδομένων του Google Landmarks Dataset v2. Αυτό ενδέχεται να έχει ως αποτέλεσμα απώλεια πληροφοριών, ιδίως για λιγότερο συχνές κατηγορίες, και να επηρεάσει τη γενικότητα και την ικανότητα του μοντέλου να αναγνωρίζει νέα, άγνωστα μνημεία.
- **Εξάρτηση από το CLIP:**
  Το μοντέλο βασίζεται στις προεκπαιδευμένες ικανότητες του CLIP. Ενώ αυτό παρέχει ισχυρές αρχικές αναπαραστάσεις, περιορίζει τη δυνατότητα fine-tuning σε πολύ εξειδικευμένες εργασίες, καθώς το backbone παραμένει σχετικά αμετάβλητο. Η προσαρμογή γίνεται κυρίως μέσω του Linear Classifier, με περιορισμένο βαθμό ρύθμισης των βαθιών στρώσεων του CLIP για πολύ ειδικές περιπτώσεις.

Συμπέρασμα:

Παρά τους περιορισμούς που προκύπτουν από το μέγεθος του dataset και την εξάρτηση από τις προεκπαιδευμένες δυνατότητες του CLIP, η προσέγγιση αυτή αποδεικνύει τη δύναμη και την ευελιξία της τεχνολογίας. Με αποτελεσματικό φιλτράρισμα, προσεκτική εκπαίδευση και χρήση του CLIP σε συνδυασμό με έναν απλό Linear Classifier, το σύστημα επιτυγχάνει εξαιρετική ακρίβεια στην αναγνώριση μνημείων, καθιστώντας το έτοιμο για εφαρμογή σε πραγματικές συνθήκες.

7. Πρόβλημα: Μη Ικανοποιητική Πρόβλεψη των Landmarks από το Μοντέλο

Τι Παρατηρούμε;

Παρά τα υψηλά ποσοστά ακρίβειας που εμφανίστηκαν κατά τη διάρκεια της εκπαίδευσης, το μοντέλο δυσκολεύεται να προβλέψει σωστά τα landmarks όταν δοκιμάζουμε νέες εικόνες ή ακόμη και εικόνες που υπήρχαν στο training set. Οι προβλέψεις που λαμβάνονται συχνά δεν αντιστοιχούν με τα αναμενόμενα

αποτελέσματα, γεγονός που εγείρει ανησυχίες σχετικά με την ικανότητα γενίκευσης του μοντέλου σε νέα δεδομένα.

Πιθανοί Παράγοντες Πρόκλησης

- **Validation Split μη Αντιπροσωπευτικό:**
  Το validation set μπορεί να περιέχει εύκολες ή διπλότυπες εικόνες, οι οποίες οδηγούν σε τεχνητά υψηλές μετρικές ακρίβειας. Αυτό ενδέχεται να μην αντικατοπτρίζει την πραγματική απόδοση του μοντέλου σε νέα, αδιάγνωστα δεδομένα.
- **Underfitting ή Overfitting:**
  - **Underfitting:** Αν υπάρχουν πολύ λίγες εικόνες ανά κατηγορία ή αν η εκπαίδευση δεν διήρκεσε αρκετά epochs, το μοντέλο δεν έχει μάθει επαρκώς να διακρίνει τα χαρακτηριστικά των διαφόρων μνημείων.
  - **Overfitting:** Αν το μοντέλο έχει εκπαιδευτεί υπερβολικά στον συγκεκριμένο χώρο του training set χωρίς επαρκή γενίκευση, μπορεί να μην αποδίδει καλά σε νέες συνθήκες, όπως διαφορετικές γωνίες λήψης ή φωτισμός.
- **Διαφορετικά Transforms στο Training και στο Inference:**
  Αν δεν χρησιμοποιούνται οι ίδιες διαδικασίες επεξεργασίας εικόνων (όπως το ίδιο resize και normalization) τόσο κατά την εκπαίδευση όσο και κατά το inference, μπορεί να προκύψουν ανακρίβειες στις προβλέψεις.
- **Pipeline Mismatch:**
  - **Φόρτωση λάθους checkpoint:** Η χρήση λανθασμένου μοντέλου ή checkpoint κατά το inference μπορεί να οδηγήσει σε ανακριβείς προβλέψεις.
  - **Εσφαλμένη κατάσταση λειτουργίας του μοντέλου:** Αν το μοντέλο δεν τίθεται σε eval() mode κατά το inference, ενεργοποιούνται μηχανισμοί όπως το dropout, που μπορεί να αλλοιώσουν τα αποτελέσματα.
  - **Αντιστοίχιση ετικετών (label mapping):** Εάν το label mapping δεν ευθυγραμμίζεται σωστά με τις κλάσεις του εκπαιδευμένου μοντέλου, οι προβλέψεις μπορεί να μεταφράζονται λανθασμένα.
- **Απουσία Επαρκούς Test Set:**
  Χωρίς ένα ξεχωριστό και ανεξάρτητο test set, τα αποτελέσματα που βασίζονται αποκλειστικά στο validation μπορεί να είναι αισιόδοξα και να μην αντικατοπτρίζουν την πραγματική απόδοση του μοντέλου σε νέα δεδομένα.

Πιθανοί Τρόποι Βελτίωσης

- **Δημιουργία Επιπλέον Test Set:**
  Συλλογή ενός εντελώς ξεχωριστού test set για ακριβέστερη αξιολόγηση της απόδοσης του μοντέλου σε αδυναμία ειδικά νέες εικόνες.
- **Έλεγχος Data Leakage:**
  Διασφάλιση ότι το validation set δεν περιέχει εικόνες που επαναλαμβάνονται στο training set. Αυτό μπορεί να γίνει μέσω προσεκτικού split ή ελέγχου διπλοτύπων.
- **Εναρμόνιση των Transforms:**
  Επιβεβαίωση ότι οι διαδικασίες προεπεξεργασίας εικόνων (resize, normalization κτλ.) είναι ίδια κατά την εκπαίδευση και το inference, για να διατηρηθεί η συνέπεια των εισόδων.
- **Ενίσχυση Data Augmentation:**
  Εφαρμογή πιο επιθετικών τεχνικών augmentation (π.χ. random crop, flip, color jitter) για αύξηση της ποικιλίας των δεδομένων κατά την εκπαίδευση, προκειμένου να βελτιωθεί η γενίκευση του μοντέλου.
- **Ρύθμιση Hyperparameters και Επιπλέον Epochs:**
  Δοκιμή μεγαλύτερου αριθμού epochs, προσαρμογή του learning rate, καθώς και χρήση τεχνικών όπως freeze/unfreeze των layers του CLIP για σταδιακή εκπαίδευση. Αυτές οι μέθοδοι μπορούν να βοηθήσουν στη βελτίωση των προβλέψεων αν το μοντέλο δεν έχει καταφέρει να μάθει επαρκώς.
- **Παρακολούθηση των Metrics:**
  Εξέταση της εξέλιξης της ακρίβειας στο training και validation set για να διαπιστώσουμε αν το μοντέλο μαθαίνει πραγματικά ή απλώς αποστηθίζει δεδομένα, και έτσι να αναπροσαρμόζουμε τη στρατηγική εκπαίδευσης.

8. Παραρτήματα

Τεχνικές Λεπτομέρειες

Δημιουργία του label_mapping.json:
Το αρχείο label_mapping.json συνδέει τα encoded labels (π.χ. 0, 1, 2, ...) με τα αντίστοιχα landmark IDs και φιλικά ονόματα, όπως "Eiffel Tower". Η διαδικασία δημιουργίας του πραγματοποιήθηκε ως εξής:

1. **Ανάγνωση του final_filtered_train_encoded.csv:**
   Από αυτό το αρχείο αντιστοιχίστηκαν τα encoded_label με τα αντίστοιχα landmark_id.

2. **Χρήση του train_label_to_category.csv:**
   Αυτό το αρχείο παρέχει URLs κατηγοριών για κάθε landmark_id, όπως το "https://commons.wikimedia.org/wiki/Category:Eiffel_Tower".
3. **Εξαγωγή φιλικών ονομάτων:**
   Από το URL αφαιρούνται τα "Category:" και οι χαρακτήρες "_" αντικαθίστανται με κενά, παράγοντας φιλικά ονόματα όπως "Eiffel Tower".
4. **Αποθήκευση σε JSON μορφή:**
   Το τελικό dictionary αποθηκεύτηκε ως label_mapping.json για χρήση κατά το inference.

```
import pandas as pd
import csv
import json

# Φόρτωση δεδομένων
encoded_csv_path = "final_filtered_train_encoded.csv"
train_label_to_category = "train_label_to_category.csv"

# Δημιουργία dictionary για τα encoded labels
df_encoded = pd.read_csv(encoded_csv_path)
label2lid = {row['encoded_label']: row['landmark_id'] for _,
row in df_encoded.iterrows()}

# Δημιουργία dictionary για τα φιλικά ονόματα
lid2category = {}
with open(train_label_to_category, "r") as f:
     reader = csv.DictReader(f)
     for line in reader:
     lid2category[int(line["landmark_id"])] =
line["category"]

# Τελικό mapping
label_mapping = {
     enc: lid2category.get(lid,
"Unknown").replace("Category:", "").replace("_", " ")
     for enc, lid in label2lid.items()
}

# Αποθήκευση σε JSON
```

```
with open("label_mapping.json", "w") as f:
     json.dump(label_mapping, f, ensure_ascii=False,
indent=2)
```

Gradient Clipping: Για να αποφύγουμε το πρόβλημα των **exploding gradients** κατά την εκπαίδευση, εφαρμόσαμε gradient clipping με max_norm 1.0. Αυτό περιορίζει τα gradients έτσι ώστε να μην υπερβαίνουν ένα συγκεκριμένο όριο, εξασφαλίζοντας σταθερότητα στην εκπαίδευση.

```
torch.nn.utils.clip_grad_norm_(model.parameters(),
max_norm=1.0)
```

Πηγές και Εργαλεία

- **Dataset:** Google Landmarks Dataset v2.
- **Βιβλιοθήκες & Frameworks:**
  - PyTorch για την εκπαίδευση του μοντέλου.
  - Streamlit για την υλοποίηση του UI.
- **Βιβλιογραφία:**
  - Radford, A. et al., "Learning Transferable Visual Models From Natural Language Supervision."
  - Official CLIP GitHub Repository: https://github.com/openai/CLIP.
- **Εργαλεία Ανάλυσης:**
  - TQDM για progress bars.
  - Matplotlib για οπτικοποίηση learning curves.

Κώδικας και Βοηθητικές Συναρτήσεις

1. Συνάρτηση για Επαλήθευση Εικόνων (Corrupted/Missing):
Αυτή η συνάρτηση ελέγχει αν οι εικόνες υπάρχουν και δεν είναι κατεστραμμένες.

```
import os
from PIL import Image

def verify_images(image_paths):
```

```
    missing_images = []
    corrupted_images = []

    for path in image_paths:
    if not os.path.exists(path):
        missing_images.append(path)
    else:
        try:
            with Image.open(path) as img:
                img.verify()
        except Exception:
            corrupted_images.append(path)

    return missing_images, corrupted_images
```

2. Collate Function για DataLoader:

Η συνάρτηση αυτή φιλτράρει κατεστραμμένα δεδομένα από κάθε batch, εξασφαλίζοντας ότι μόνο έγκυρα δείγματα περνάνε στη συνέχεια.

```
def collate_fn(batch):
    batch = [item for item in batch if item[0] is not None
and item[1] is not None]
    if not batch:
    return torch.empty(0, 3, 224, 224), torch.empty(0,
dtype=torch.long)
    images, labels = zip(*batch)
    return torch.stack(images), torch.tensor(labels)
```

3. Early Stopping Mechanism:

Αυτός ο μηχανισμός παρακολουθεί το validation loss και σταματά την εκπαίδευση όταν δεν σημειώνεται σημαντική βελτίωση για προκαθορισμένο αριθμό εποχών, αποτρέποντας το overfitting.

```
class EarlyStopping:
    def __init__(self, patience=5, min_delta=1e-4):
    self.patience = patience
    self.min_delta = min_delta
```

```
    self.counter = 0
    self.best_score = None
    self.early_stop = False

    def __call__(self, current_val_loss):
    if self.best_score is None:
         self.best_score = current_val_loss
         return

    improvement = self.best_score - current_val_loss
    if improvement < self.min_delta:
         self.counter += 1
         if self.counter >= self.patience:
              self.early_stop = True
    else:
         self.best_score = current_val_loss
         self.counter = 0
```