



UNIVERSITÀ DEGLI STUDI DI CATANIA
DIPARTIMENTO DI MATEMATICA E INFORMATICA
CORSO DI LAUREA TRIENNALE IN INFORMATICA

Kevin Speranza

Egocentric Videos Understanding with Active Memory

RELAZIONE PROGETTO FINALE

Relatore: Francesco Ragusa

Anno Accademico 2024 - 2025

Abstract

In questa tesi si analizza l'uso dei video egocentrici per lo studio del comportamento umano in scenari operativi, con particolare attenzione alla gestione delle interazioni tra mani e oggetti. Viene presentato AMEGO [1], un framework semantic-free per la strutturazione di una memoria capace di fornire informazioni basate su dati visuali-temporali.

Il lavoro prevede la realizzazione di un benchmark in un contesto industriale, utilizzando il dataset *ENIGMA-51* [2], opportunamente adattato per valutare le prestazioni del modello in un dominio differente rispetto a quello originariamente considerato [3]. Le valutazioni mostrano limiti nella distinzione tra oggetti visivamente simili, sottolineando la necessità di strategie più robuste per il *clustering* e l'organizzazione degli oggetti.

Nonostante alcune limitazioni, AMEGO costituisce una base solida per l'analisi dei video egocentrici, offrendo diversi spunti per sviluppi futuri volti a potenziarne le prestazioni anche in contesti differenti.

Indice

1	Introduzione	3
2	Lavori correlati	5
2.1	Long video understanding benchmarks	5
2.2	Structured video representations	6
2.3	Video summarization	7
3	Metodo - AMEGO	8
3.1	Costruzione della memoria	8
3.2	Pseudocodici	14
3.3	AMB - Active Memories Benchmark	17
4	Dataset	21
4.1	EPIC-KITCHENS	21
4.2	ENIGMA-51	25
5	Esperimenti	29
5.1	Setup ambiente	30
5.2	AMB su ENIGMA-51	31
5.3	Analisi	36
6	Risultati	39
6.1	EPIC-KITCHENS	39
6.2	ENIGMA-51	40
	Conclusione	46
	Bibliografia	48

Capitolo 1

Introduzione

Negli ultimi anni i dispositivi indossabili per la registrazione di video in prima persona hanno conosciuto una diffusione sempre più ampia. Strumenti come *smart glasses*, *body cameras*, *action cameras* hanno reso possibile la cattura di flussi video continui dal punto di vista dell'utilizzatore, dando origine a quello che viene comunemente definito come *egocentric video*. Ciò che li rende peculiari è la loro capacità di catturare dettagli e prospettive uniche, fornendo una visione diretta dell'attività di chi li indossa.



Figura 1.1: Dispositivi indossabili per la cattura di video egocentrici

L'adozione crescente di tali dispositivi è stata favorita dalla loro versatilità: da un lato vengono utilizzati per scopi ricreativi e per la condivisione di esperienze personali, dall'altro trovano applicazione in contesti professionali e industriali, dove consentono di documentare procedure complesse e migliorare i processi di formazione e supervisione.

Il principale ostacolo all'analisi di questi contenuti risiede nella loro natura non strutturata. I video in prima persona possono essere considerati come veri e propri flussi di coscienza visivi: lunghi, frammentati, privi di un'organizzazione narrativa chiara e difficili da interpretare. La presenza di movimenti rapidi della fotocamera, variazioni di illuminazione e interazioni simultanee con più oggetti rende complicata l'estrazione di significato.

Un’annotazione manuale completa non è praticabile, sia per la mole di dati prodotta sia per la complessità dei contenuti.

Da questa problematica emerge la necessità di costruire una “memoria artificiale” in grado di trasformare i video egocentrici in una rappresentazione più organizzata e interpretabile. In questa tesi analizzeremo innanzitutto i principali contributi presenti in letteratura, per poi concentrarci su AMEGO[1], un sistema sviluppato per strutturare e rendere interrogabili i video egocentrici e attualmente considerato stato dell’arte nel suo ambito [1].

Durante la fase sperimentale, valuteremo AMEGO in un contesto diverso da quello in cui era stato testato. Nel lavoro originale è stato valutato sul dataset *EPIC KITCHENS* [3], costituito da video ambientati in cucine domestiche. In questa tesi, invece, l’analisi si concentra sul dataset *ENIGMA-51* [2], acquisito in contesti industriali, in cui diversi operatori hanno seguito procedure guidate per eseguire attività di riparazione di quadri elettrici. La differenza tra i due domini rende lo studio particolarmente interessante, in quanto consente di valutare la capacità di AMEGO di generalizzare a contesti applicativi mai visti.

In ambito industriale, l’analisi dei video egocentrici è fondamentale per ottimizzare vari aspetti operativi. Garantire ad esempio che le operazioni vengano eseguite nell’ordine corretto migliora i flussi produttivi e riduce il rischio di errori umani. Inoltre, la capacità di riconoscere quali strumenti vengono utilizzati contemporaneamente ad altri oggetti, soprattutto se potenzialmente pericolosi, contribuisce a migliorare la sicurezza dei lavoratori. Quest’ultimo tema prende il nome di **concurrency**. Sarà di centrale importanza nella fase sperimentale.

Capitolo 2

Lavori correlati

2.1 Long video understanding benchmarks

La comprensione di video con una lunga durata richiede l'analisi e l'interpretazione di contenuti visivi che possono estendersi per diversi minuti o addirittura ore. Questi video richiedono metodi capaci di catturare sequenze temporali complesse e le interazioni tra più oggetti e persone, il che rende la loro gestione particolarmente impegnativa dal punto di vista computazionale.

Per affrontare questa sfida sono stati sviluppati benchmark specifici che mettono alla prova la capacità dei modelli su questo tipo di task. Tra i più rilevanti troviamo EgoSchema [4], un dataset con video egocentrici della durata massima di circa tre minuti, progettato per analizzare azioni quotidiane articolate in più passaggi. I video includono azioni semplici come manipolare strumenti e materiali di uso quotidiano. I modelli devono quindi avere una coerenza spaziotemporale dei vari movimenti, riconoscere pattern ricorrenti, e inferire correttamente le relazioni tra: mani, oggetti e azioni.

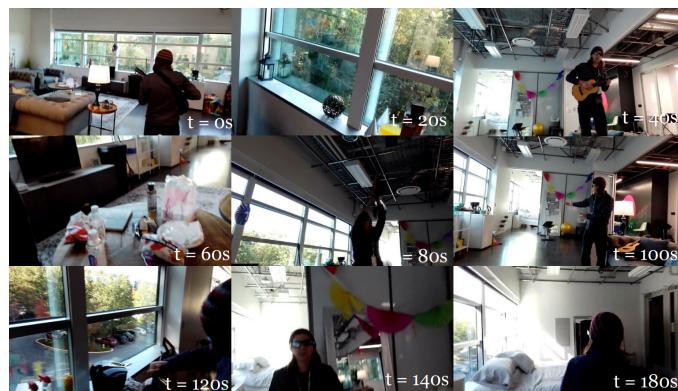


Figura 2.1: EgoSchema Dataset

ReST [5] propone invece scenari industriali e scientifici più complessi, con video più lunghi in cui interagiscono simultaneamente più strumenti, oggetti e operatori.

Diversi approcci sono stati sviluppati: alcuni trattano il problema come un task di *natural language question answering*, generando prima dei sottotitoli o descrizioni automatiche del video e utilizzando LLM per rispondere a domande specifiche [6, 7, 8, 9, 10, 11]. Altri approcci integrano direttamente LLM con un encoder video, sfruttando le capacità di comprensione e generazione dei modelli linguistici per elaborare sequenze visive estese in maniera coerente [12, 13, 14, 15].

2.2 Structured video representations

Con *rappresentazione strutturata* si intende l'insieme di tecniche volte a organizzare un video non come una semplice sequenza di frame, ma come una struttura semantica in grado di esplicitare le relazioni tra gli elementi presenti nella scena. Questo approccio consente di passare da una descrizione puramente visiva a una rappresentazione schematica e strutturata, che rende possibile interrogare i video in maniera più efficace, permettendo di estrarre informazioni mirate.

Un filone centrale della ricerca si è concentrato sullo studio delle relazioni contestuali, investigando in particolare i legami tra oggetti e attori [16, 17, 18, 19, 20, 21, 22, 23]. Parallelamente, sono stati proposti modelli basati su grafi per rappresentare le dipendenze tra azioni, al fine di catturare la dimensione temporale e causale dei comportamenti.

Lavori invece come UnweaveNet [24] propongono di raggruppare i video in *activity threads*, ossia insiemi di clip collegati logicamente che consentono di separare e ricostruire le diverse “storie” di attività intrecciate all’interno di una sequenza più lunga. Un altro contributo rilevante è l’introduzione dei cosiddetti *egocentric scene graphs* [25], a cui hanno partecipato i professori Antonino Furnari e Giovanni Maria Farinella del Dipartimento di Matematica e Informatica dell’Università di Catania. Queste strutture sono progettate per rappresentare in modo esplicito le interazioni tra il soggetto che indossa la videocamera e gli oggetti presenti nell’ambiente.

Nonostante questi progressi, tali approcci risultano ancora limitati, poiché tendono a catturare soltanto alcuni aspetti delle attività. In particolare, faticano a integrare in un unico modello le molteplici dimensioni tipiche dei video egocentrici: le interazioni con gli oggetti, i luoghi chiave in cui esse avvengono e le interdipendenze tra questi elementi. Questa mancanza di completezza riduce la capacità di ottenere rappresentazioni realmente efficaci.

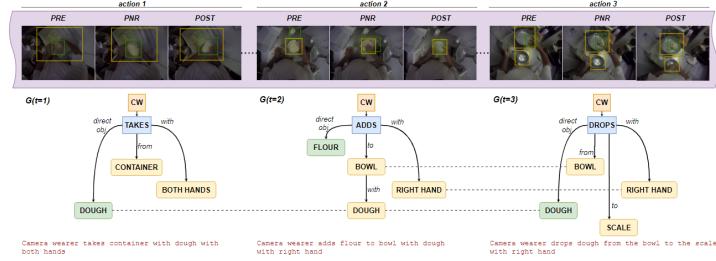


Figura 2.2: Egocentric scene graph. I nodi rappresentano attori e oggetti, mentre gli archi indicano le relazioni tra di essi.

2.3 Video summarization

Il riassunto del video ha come obiettivo la generazione di una versione ridotta di un video, tipicamente attraverso l'estrazione di *key frames* che ne catturino i momenti salienti. Gli approcci proposti in letteratura variano in funzione degli elementi ritenuti rilevanti per la sintesi: alcuni si focalizzano sulla presenza e sul ruolo delle persone o degli oggetti all'interno della scena [26], altri privilegiano la rilevazione di eventi significativi [27], mentre ulteriori metodi considerano anche caratteristiche estetiche dei fotogrammi chiave per selezionare i contenuti più rappresentativi [28].

Accanto a questi, sono stati introdotti approcci in grado di generare i riassunti in modalità *online*, ossia durante la riproduzione del flusso video, consentendo una sintesi in tempo reale [29, 30]. Tuttavia, tali tecniche non costruiscono una rappresentazione strutturata del video e risultano spesso sensibili al rumore prodotto dai modelli di rilevamento, mancando di un'efficace integrazione della dimensione temporale.

Un contributo particolarmente rilevante in questa direzione è rappresentato da [31], che permette di interrogare i contenuti lungo diverse dimensioni, anche combinandoli attraverso operatori booleani. Ciononostante, il metodo rimane in gran parte vincolato al riconoscimento di attrazioni predefinite e di oggetti visivamente distinti, risultando meno efficace in scenari affollati e caotici.

Capitolo 3

Metodo - AMEGO

AMEGO, acronimo di *Active Memory of the EGOCentric video*, è concepito per trasformare un video egocentrico lungo e non strutturato in una memoria capace di descrivere in modo completo le interazioni del soggetto con oggetti e luoghi. Allo stesso tempo, può essere interrogato per recuperare i segmenti temporali in cui un oggetto è stato utilizzato, una location è stata visitata, o entrambe le condizioni si sono verificate contemporaneamente.

Un aspetto cruciale che distingue AMEGO da altri approcci riguarda la sua natura *semantic-free*. Gli oggetti e le location non vengono legati a una tassonomia fissa di etichette o a un vocabolario prestabilito. Essi vengono invece rappresentati direttamente sulla base delle caratteristiche visive, consentendo così una distinzione più fine e dettagliata tra le diverse istanze. Questo approccio permette al sistema di adattarsi a contesti nuovi senza la necessità di ridefinire un insieme di categorie predefinite.

3.1 Costruzione della memoria

Dato un video egocentrico \mathcal{V} , esso viene scomposto in due elementi fondamentali:

- **Hand-Object Interaction (HOI) tracklets:** ciascun HOI tracklet¹ descrive in maniera spaziotemporale un oggetto che interagisce in modo consistente con almeno una mano del soggetto. Ogni tracklet è caratterizzato da bounding boxes² e dalle corrispondenti feature visive³.

¹**Tracklet:** sequenza di bounding box che identifica in modo coerente la traiettoria o l'interazione di un oggetto nel tempo.

²**Bounding box:** regione rettangolare che delimita un oggetto in un singolo frame del video.

³**Feature visive:** rappresentazioni numeriche delle proprietà visive di un oggetto

- **Location segments:** ogni elemento corrisponde a un intervallo temporale in cui il soggetto si trova in un determinato luogo e vi svolge interazioni. L'interesse è focalizzato sulle cosiddette *activity-centric-zones*, ossia i luoghi in cui avvengono le principali interazioni con gli oggetti.

Combinando gli *HOI tracklets* con i *Location segments* si ottiene una memoria strutturata in grado di eseguire i compiti discussi in precedenza.

La memoria viene definita come:

$$\mathcal{E} = \{\mathcal{O}, \mathcal{L}\}$$

dove:

- \mathcal{E} : AMEGO
- \mathcal{O} : insieme di HOI tracklets
- \mathcal{L} : insieme dei Location Segments.

Questa memoria viene costruita *online*, eliminando la necessità di riprocessare continuamente informazioni passate.

Object interaction tracklets

Gli *HOI tracklets*, indicati con \mathcal{O} rappresentano sequenze di interazioni tra le mani del soggetto e gli oggetti presenti nel video. Formalmente, possiamo definire l'insieme degli HOI tracklets come:

$$\mathcal{O} = \{o_1, o_2, \dots, o_n\}$$

dove ciascun tracklet $o_i \in \mathcal{O}$ è una tupla:

$$o_i = (t_s, t_e, b_t, h, \text{id})$$

con:

- t_s : istante di inizio dell'interazione
- t_e : istante di fine dell'interazione
- b_t : sequenza di bounding box che raffigurano l'oggetto
- h : lato della mano che compie l'interazione (sinistra o destra)
- id : identificatore dell'istanza dell'oggetto associato al tracklet

La costruzione della memoria \mathcal{O} avviene in maniera iterativa, processando il video frame per frame tramite una pipeline composta da tre fasi principali:

1. **Initialization:** individuazione dei possibili nuovi HOI tracklets.
2. **Updating:** aggiornamento dei tracklets attivi⁴, corrispondenti alle interazioni in corso.
3. **Assignment and storing:** i tracklets terminati⁵ vengono archiviati nella memoria \mathcal{E} e viene assegnata loro l'istanza oggetto corrispondente.

Inizializzazione

La prima fase consiste nell'individuazione dei nuovi *HOI tracklets*. Per questo utilizziamo un detector di *hand-object-interaction class-agnostic*⁶ [32], che fornisce insiemi di bounding box attive per oggetti e mani, denotati rispettivamente come \mathcal{B}_t^o e \mathcal{B}_t^h .

Un nuovo *HOI tracklet* o_i viene inizializzato per ciascuna nuova hand-object-interaction rilevata. Ogni tracklet è definito come una sequenza di almeno s_o bounding box che mostrano un forte sovrapposizione spaziale all'interno di una finestra temporale di w_s frame.

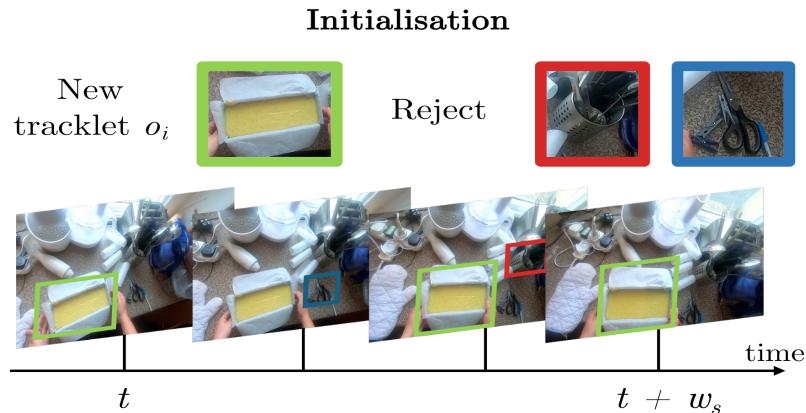


Figura 3.1: Fase di inizializzazione

Questo processo di filtraggio consente di ridurre il rumore generato dall'applicazione indipendente del rilevatore su ciascun frame. Considerando la

⁴**Tracklet attivi:** tracklets che stanno effettivamente registrando un'interazione in corso tra la mano del soggetto e l'oggetto

⁵**Tracklet terminato:** l'azione per cui veniva considerato attivo è terminata

⁶**class-agnostic detector:** non fa distinzione tra classi predefinite di oggetti, ma identifica interazioni tra mani e oggetti basandosi su caratteristiche visive generiche

durata naturale delle interazioni mano-oggetto, è possibile identificare in modo affidabile i nuovi tracklets attivi, garantendo coerenza spaziale e temporale nelle rilevazioni.

Il tracklet o_i viene ora considerato *attivo* e aggiunto alla memoria \mathcal{O} .

Per ciascun frame successivo, calcoliamo l'*Intersection over Union (IoU)*⁷ tra i bounding box degli oggetti che interagiscono con la stessa mano. I bounding box che superano una soglia θ vengono assegnati al tracklet o_i . Se non è possibile assegnare nuovi bounding box al tracklet, questo viene considerato *terminato*.

Updating

Questa fase mira a catturare l'intera durata dell'interazione e contemporaneamente a seguire tutte le occorrenze spaziali dell'oggetto.

Sebbene i rilevatori di HOI a livello di singolo frame siano sufficienti per identificare nuove interazioni, essi non sono in grado di estendere in modo affidabile i tracklets quando mani od oggetti escono dal campo visivo. Per questo motivo, viene utilizzato un *single-object tracker (SOT)*⁸ [33].

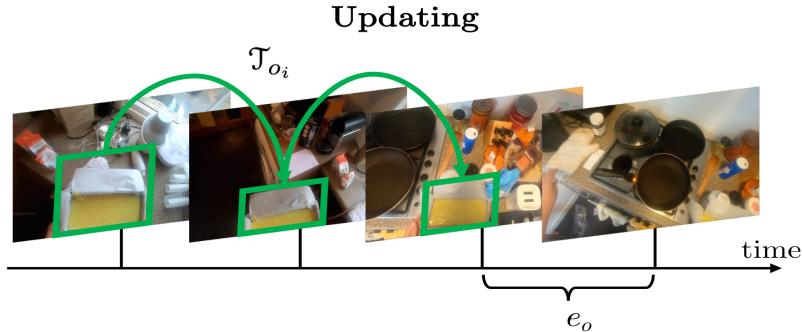


Figura 3.2: Fase di updating

Per ogni tracklet attivo o_i viene inizializzato un SOT. Consideriamo il tracklet o_i terminato se non sono presenti rilevazioni associate \mathcal{B}^o per e_o frame consecutivi, mentre la mano h rimane visibile, in quanto quando la mano esce dal campo visivo è probabile che stia ancora tenendo l'oggetto.

L'output del SOT produce un track τ_{o_i} che segue la posizione dell'oggetto, ma non contiene informazioni sull'interazione stessa. A questo punto, o_i

⁷**Intersection over Union (IoU):** misura di sovrapposizione tra due bounding box, calcolata come il rapporto tra l'area di intersezione e l'area di unione dei due rettangoli.

⁸**Single-Object Tracker (SOT):** permette di seguire un singolo oggetto nel tempo, stimando la posizione frame per frame anche in assenza di rilevazioni dirette.

combina le informazioni relative alla durata temporale (start ed end time) e ai bounding box spaziali dell'oggetto attivo, sfruttando sia la rilevazione HOI a livello di frame sia il tracciamento SOT.

Assignment and storing

Come definito in precedenza, ogni HOI tracklet o_i deve essere associato a una specifica istanza di oggetto. Per farlo, confrontiamo o_i con le istanze già presenti nella memoria.

In particolare, dato l'insieme di HOI tracklets già memorizzati al tempo t , denotato \mathcal{O}_t , e l'insieme dei tracciamenti SOT osservati nello stesso istante, τ_t , verifichiamo se o_i possa essere associato a un'istanza esistente o se sia necessario crearne una nuova. Per effettuare il confronto, calcoliamo prima le feature visive di o_i :

$$f(o_i) = \frac{1}{|\mathcal{V}_{o_i}|} \sum_{k \in \mathcal{V}_{o_i}} \gamma(k, b_k^o)$$

dove:

- \mathcal{V}_{o_i} è l'insieme dei frame associati a o_i ,
- b_k^o è il bounding box relativo al frame k ,
- γ visual-feature-extractor (nel caso di AMEGO: DINOV2 [34]).

Per effettuare il *matching*, utilizziamo un approccio di clustering online basato su $f(o_i)$. La similarità tra o_i e una specifica istanza di oggetto id_j viene calcolata come:

$$s(o_i, id_j) = \frac{1}{|\mathcal{O}_t \in id_j|} \sum_{\mathcal{O}_t \in id_j} \langle f_{\mathcal{O}_t}, f_{o_i} \rangle$$

dove $\langle \cdot, \cdot \rangle$ indica la *cosine similarity* e $\mathcal{O}_t \in id_j$ è l'insieme dei tracklets associati all'istanza id_j .

Assegniamo o_i all'istanza id_j^* che massimizza la similarità e che supera una soglia θ (diversa dalla soglia utilizzata per l'IoU). Se un tracker in τ_t si sovrappone significativamente con o_i e la sua confidenza è maggiore della similarità massima, allora o_i viene assegnato all'istanza del tracker. Altrimenti, viene assegnato a id_j^* . Qualora la similarità massima risultasse inferiore alla soglia, viene creata una nuova istanza per o_i .

Al termine di questa fase, le feature $f(o_i)$ e l'istanza assegnata vengono associate al tracklet o_i e memorizzate nella memoria \mathcal{E}

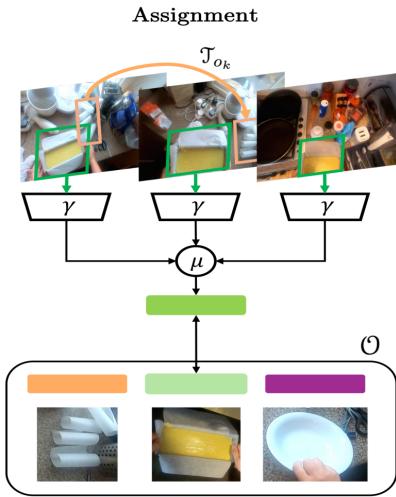


Figura 3.3: Fase di assegnamento istanza

Location segments

Definiamo l'insieme dei *Location segments* \mathcal{L} come gli intervalli temporali durante i quali il soggetto svolge interazioni in *zone di attività principali*.

Poiché un soggetto può interagire con più oggetti simultaneamente ma può trovarsi in un solo punto alla volta, ogni segmento $l_i \in \mathcal{L}$ viene modelizzato come un intervallo temporale che corrisponde all'inizio e alla fine di un'interazione in quella specifica zona.

Analogamente agli *HOI tracklets*, l'insieme \mathcal{L} viene costruito online.

Si seguono due fasi principali: Temporal segmentation e Assignment and storing

Temporal segmentation

Dati i frame \mathcal{V}_t e le rilevazioni delle mani B_t^h , verifichiamo se la mano sta interagendo con un oggetto mentre si trova in una location. Per farlo, calcoliamo l'*optical flow*⁹ tra \mathcal{V}_{t-1} e \mathcal{V}_t e controlliamo la presenza di mani assicurandoci che $|B_t^h| > 0$.

Possiamo quindi determinare se il soggetto sta svolgendo un compito considerando le seguenti condizioni:

1. l'*optical flow* ha norma¹⁰ bassa;

⁹**Optical flow:** rappresenta il campo di movimento apparente dei pixel tra due frame consecutivi di un video, indicando la direzione e la velocità dello spostamento

¹⁰**Norma:** grandezza che rappresenta l'intensità complessiva di un vettore, nel nostro caso l'*optical flow*

2. è rilevata almeno una mano.

Questi criteri permettono di stabilire se il soggetto ha fatto una pausa (basso *optical flow*) ed è attivamente coinvolto nella scena (mano rilevata).

Analogamente agli HOI, applichiamo un filtraggio temporale, per cui un *Location segment* l_j è considerato attivo solo se entrambe le condizioni sono verificate per un numero consecutivo di frame s_l . Il segmento l_j viene terminato quando osserviamo un numero consecutivo di frame e_l in cui la norma dell'*optical flow* supera la soglia o non sono presenti mani rilevate.

Assignment and storing

Come per gli HOI, dobbiamo assegnare un'istanza alle location l_j definite temporaneamente. Agiamo analogamente, utilizzando però un *visual-feature-extractor* differente, σ (SWAG)[35].

Una volta ottenute le visual-feature g_{l_j} , calcoliamo la similarità tra tutte le istanze di location già presenti e assegniamo a l_j l'id che massimizza questa similarità, a condizione che superi una soglia prestabilita τ . Se la soglia non viene superata, viene creata una nuova istanza.

$$s(l_j, id_j) = \frac{1}{|\mathcal{L}_t \in id_j|} \sum_{\mathcal{L}_t \in id_j} \langle g_{\mathcal{L}_t}, g_{l_j} \rangle$$

Al termine di queste fasi, assegniamo $g(l_j)$ e l'istanza correlata nella nostra memoria \mathcal{E} .

3.2 Pseudocodici

Di seguito vengono riportati i pseudocodici relativi alla costruzione della pipeline per la generazione degli elementi della memoria discussi nei paragrafi precedenti.

Object interactions

Algorithm 1 Object interactions pipeline

```

1: Input:
2:   Frames  $\{\mathcal{V}_t\}$ 
3:   HOI detector  $\mathcal{D}$ 
4:   SOT tracker  $\mathcal{J}$ 
5:   Similarity threshold  $\theta$ 
6: Output:
7:   Set of hand-object interaction tracklets  $\mathcal{O}$ 
8: for each frame  $\mathcal{V}_t$  do
9:    $\mathcal{B}_t^o, \mathcal{B}_t^h \leftarrow \mathcal{D}(\mathcal{V}_t)$  (Detect hands and objects)
10:  for each detection  $(b^o, b^h) \in (\mathcal{B}_t^o, \mathcal{B}_t^h)$  do
11:    if new HOI11 (i.e.  $s_o$  detections in the last  $w_s$  frames) then
12:      Create new tracklet  $o_i$ 
13:      Start SOT  $\mathcal{J}_{o_i}$  for  $o_i$ 
14:    end if
15:  end for
16:  for each completed tracklet  $o_i$  do
17:    Update the detections with  $\mathcal{J}_{o_i}$ 
18:    if  $\nexists b^o \in \mathcal{B}_t^o$  matching with  $o_i$  in the last  $e_o$  frames and  $|\mathcal{B}_t^o| > 0$ 
19:    then
20:      Mark  $o_i$  as complete
21:    end if
22:  end for
23: for each completed tracklet  $o_i$  do
24:   Compute visual features  $f(o_i)$ 
25:   Compute similarity  $s(o_i, id_j)$  with existing instances in  $\mathcal{O}$ 
26:   if maximum similarity  $> \theta$  then
27:     Assign  $o_i$  to best matching instance  $id_j$ 
28:   else
29:     Create new instance for  $o_i$ 
30:   end if
31:   Store  $o_i$  in  $\mathcal{O}$ 
32: end for
33: return  $\mathcal{O}$ 
  
```

Location segments

Algorithm 2 Location Segment pipeline

```

1: Input:
2:   Frames  $\{\mathcal{V}_t\}$ 
3:   HOI detector  $\mathcal{D}$ 
4:   Similarity threshold  $\tau$ 
5: Output:
6:   Set of location segments  $\mathcal{L}$ 
7: for each frame  $\mathcal{V}_t$  do
8:    $\mathcal{B}_t^o, \mathcal{B}_t^h \leftarrow \mathcal{D}(\mathcal{V}_t)$  (Detect hands and objects)
9:   Compute optical flow OpticalFlow( $\mathcal{V}_{t-1}, \mathcal{V}_t$ )
10:  if location segment  $l_j$  is active then
11:    if high  $|\text{OpticalFlow}(\mathcal{V}_{t-1}, \mathcal{V}_t)|$  or  $|\mathcal{B}_t^h| = 0$  for  $e_l$  consecutive
     frames then
12:      Mark  $l_j$  as complete
13:    else
14:      Continue  $l_j$ 
15:    end if
16:  else
17:    if low  $|\text{OpticalFlow}(\mathcal{V}_{t-1}, \mathcal{V}_t)|$  and  $|\mathcal{B}_t^h| > 0$  for  $s_l$  consecutive
     frames then
18:      Subject is interacting, start active location segment  $l_j$ 
19:    end if
20:  end if
21: end for
22: for each completed segment  $l_j$  do
23:   Compute visual features  $g(l_j)$ 
24:   Compute similarity  $s(l_j, id_i)$  with existing instances in  $\mathcal{L}$ 
25:   if maximum similarity  $> \tau$  then
26:     Assign  $l_j$  to best matching instance  $id_i$ 
27:   else
28:     Create new instance for  $l_j$ 
29:   end if
30:   Store  $l_j$  in  $\mathcal{L}$ 
31: end for
32: return  $\mathcal{L}$ 
  
```

3.3 AMB - Active Memories Benchmark

Per studiare le interazioni dei vari elementi è stato introdotto un benchmark ad-hoc, l'*Active Memories Benchmark* (AMB).

Il benchmark comprende 20.500 query che coprono diversi livelli di ragionamento. Le query sono formulate come domande a scelta multipla.

Ad esempio, alcune domande riguardano l'utilizzo di oggetti: “Quale oggetto ho usato con [VQ]?” dove [VQ] rappresenta un ritaglio visivo dell’oggetto; altre domande chiedono “In quali location ho usato [VQ]?”

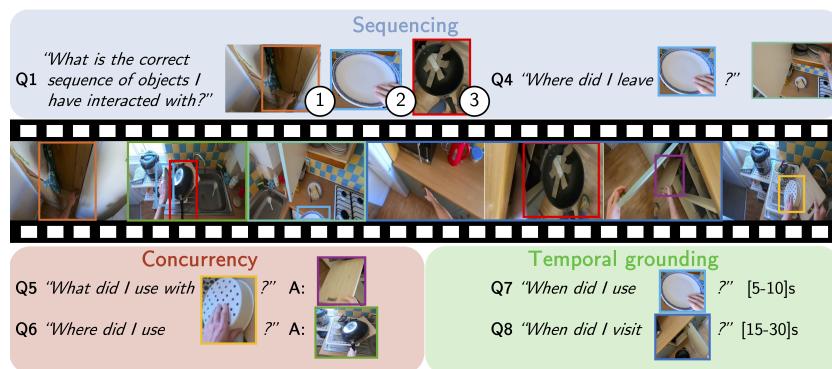


Figura 3.4: Query del benchmark AMB

Come si nota, nelle domande non compaiono i nomi degli oggetti. Ogni visual query di un oggetto (VQ), risposta visiva dell’oggetto (VA), query sulla location (LQ) o risposta sulla location (LA) è parametrizzata tramite patch visive.

Tipologie di query

Le query sono strutturate in tre macro aree:

- **Sequencing (SQ) [Q1-4]:** valutano la capacità di discriminare l’ordine temporale degli eventi. Ad esempio, il modello deve ordinare le interazioni nel tempo e identificare quale oggetto è stato utilizzato prima o dopo un altro.
- **Concurrency (CO) [Q5-6]:** valutano la capacità di catturare interazioni multiple simultanee. Ad esempio, verificare se diversi oggetti sono stati utilizzati insieme (oggetto-oggetto) o se un oggetto è stato usato in una specifica location (oggetto-location).

- **Temporal Grounding (TG) [Q7-8]:** valutano la capacità di recuperare tutti gli intervalli temporali in cui un oggetto o una location è stato coinvolto in interazioni.

Reasoning	Query	Template	Dim.	Answer
SQ	Q1	What is the correct sequence of objects I have interacted with?	O	Obj. seqs
	Q2	What did I use with the left/right hand after [VQ]?	O	Obj.
	Q3	What did I use with the left/right hand before [VQ]?	O	Obj.
	Q4	Where did I take/leave [VQ]?	O, L	Loc.
CO	Q5	What did I use with [VQ]?	O	Obj. sets
	Q6	Where did I use [VQ]?	O, L	Loc. sets
TG	Q7	When did I use [VQ]?	O	Intervals
	Q8	When did I visit [LQ]?	O, L	Intervals

Tabella 3.1: Tipologie di domande AMB con colonne colorate per tipo di Reasoning

Risposte alle Query

Inizialmente, si recupera la rappresentazione più vicina dell'oggetto o della location della query, quindi vengono applicate euristiche specifiche per ciascuna domanda. Un esempio viene fornito al seguente link: <https://gabrielegoletto.github.io/AMEGO/#querying>. Di seguito vengono spiegati i dettagli teorici delle implementazioni.

Q1: Sequenze di oggetti

Si confrontano tutti gli oggetti presenti nelle sequenze associate a ciascuna risposta, assegnando ogni patch d'immagine a un oggetto q_id tra quelli in \mathcal{O} . Successivamente, si seleziona la risposta con la sottosequenza comune più lunga calcolata tra la sequenza completa di \mathcal{O} e ciascuna delle risposte.

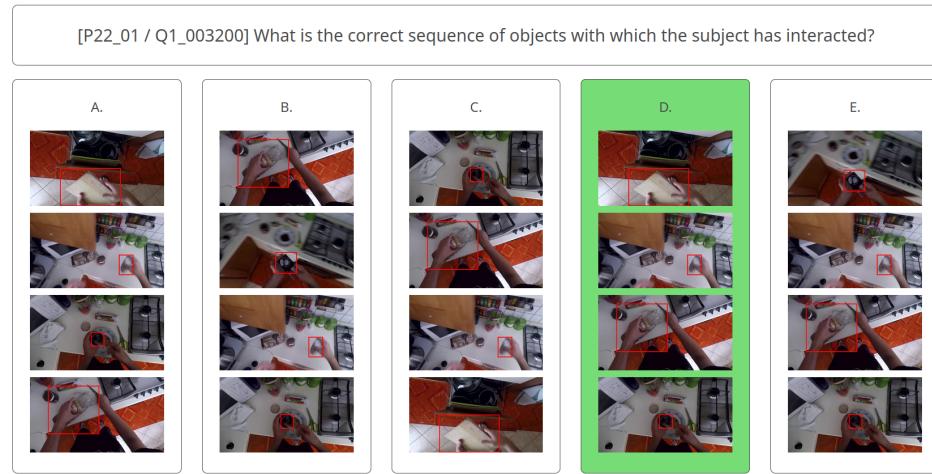


Figura 3.5: Esempio di risposta a Q1.

Q2-Q3: Oggetto prima o dopo la query

Si identifica la traccia dell'oggetto della query in \mathcal{O} basandosi su tre criteri:

- vicinanza temporale a t ;
- presenza del lato della mano indicato nella domanda;
- similarità minima di 0.6.

Una volta individuata la traccia corrispondente, per Q2 si estraе la parte successiva della traccia contenente la mano della query, mentre per Q3 si estraе la parte precedente, confrontandola con le risposte e selezionando quella con la similarità più alta.

Q4: Location iniziale/finale

Si individua l'oggetto della query in \mathcal{O} e il corrispondente q_id . Usando questo ID, si identifica il primo o l'ultimo segmento di location in cui l'oggetto appare in \mathcal{E} e si confrontano le location con le risposte. La risposta selezionata è quella con la similarità più alta.

Q5: Oggetti concorrenti

Si confronta l'oggetto della query con le tracce in \mathcal{O} per ottenere il q_id . Allo stesso modo, si estraggono gli ID degli oggetti di ciascuna risposta. La risposta selezionata è quella con il maggior numero di oggetti che coesistono temporalmente con q_id (segmenti temporali sovrapposti) in \mathcal{E} .

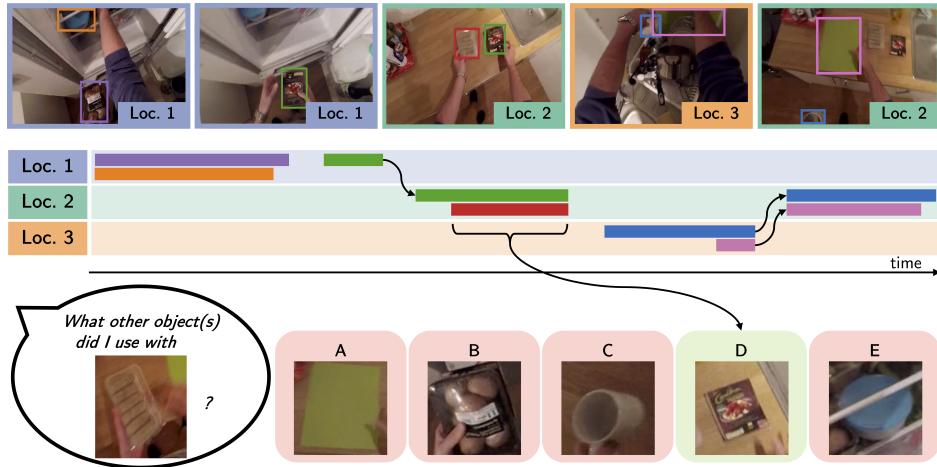


Figura 3.6: Esempio di risposta a Q5.

Q6: Location concorrenti

Analogamente a Q5, si individua q_id dell'oggetto della query e si estraggono gli ID delle location dalle risposte. La risposta selezionata è quella con il maggior numero di location che coesistono temporalmente con q_id in \mathcal{E} .

Q7-Q8: Intervalli temporali

Si confronta l'oggetto o la location della query con le tracce in \mathcal{O} o \mathcal{L} . Successivamente, si estraggono tutti gli intervalli temporali in cui l'istanza era attiva in \mathcal{E} . La risposta selezionata è quella con il più alto Intersection over Union (IoU) medio temporale.

Capitolo 4

Dataset

I dataset utilizzati in questo lavoro sono due: *EPIC-KITCHENS* [3], utilizzato nativamente da AMEGO per la costruzione del benchmark discusso in precedenza (AMB), ed *ENIGMA-51* [2], oggetto principale del lavoro di tesi e rifinito per valutare il metodo su un contesto differente.

4.1 EPIC-KITCHENS

EPIC-KITCHENS è un dataset egocentrico raccolto in contesti domestici di cucina. Contiene registrazioni video di attività quotidiane svolte da singoli individui in cucina, con annotazioni di azioni e oggetti coinvolti.

Collezione dei dati

Il dataset è stato registrato da 32 partecipanti, ciascuno dei quali ha documentato tutte le visite in cucina. Le registrazioni iniziano immediatamente prima dell'ingresso in cucina e terminano appena prima dell'uscita. I partecipanti sono soli in cucina durante le registrazioni, in modo da catturare esclusivamente attività di una singola persona.

La cattura dei dati è stata effettuata mediante una GoPro montata in testa, con un supporto regolabile per adattarsi all'altezza dei diversi soggetti e all'ambiente.

Le riprese sono in risoluzione Full HD di 1920x1080 a 59.94 fps. Alcune registrazioni hanno risoluzioni o frame rate differenti: 1% a 1280x720, 0.5% a 1920x1440, 1% a 30 fps, 1% a 48 fps e 0.2% a 90 fps.

In media, le registrazioni hanno una durata di 1.7 ore, con un massimo di 4.6 ore per soggetto.



Figura 4.1: Modalità di ripresa con GoPro montata in testa

Annotationi

La pipeline di annotazione ha avuto inizio con una prima fase di narrazione da parte dei partecipanti stessi, seguita da un processo di crowdsourcing su Amazon Mechanical Turk (AMT) [36] per la rifinitura e la validazione dei dati [3]. Le annotazioni raccolte si dividono principalmente in tre tipologie: narrazioni video, segmenti di azione e bounding box degli oggetti.

Narrazioni Video

Come primo passo, è stato chiesto ai partecipanti di guardare i propri video dopo aver completato tutte le registrazioni e di narrare a voce le azioni che avevano compiuto [3]. Le narrazioni sono state raccolte in 5 lingue diverse (inglese, italiano, spagnolo, greco e cinese), a seconda della lingua madre del partecipante [3].

Successivamente, queste registrazioni audio sono state trascritte manualmente tramite AMT, poiché i sistemi di speech-to-text automatici [37, 38, 39] si sono rivelati inefficaci a causa del lessico specifico e delle frasi incomplete [3]. I timestamp di ogni narrazione sono stati ottenuti allineando l'audio al video tramite lo strumento di sottotitolaggio automatico di YouTube. Questo processo ha portato alla raccolta di 39,596 narrazioni di azioni.

Segmenti di Azione

Le narrazioni hanno fornito una prima localizzazione temporale approssimativa delle azioni [3]. Per ottenere una segmentazione precisa, per ogni frase narrata, i tempi esatti di inizio e fine dell'azione corrispondente sono stati annotati tramite un ulteriore task su AMT [3]. In questo modo, sono stati

etichettati 39,564 segmenti di azione, con una durata media di 3.7 secondi [3].



Figura 4.2: Esempi di azioni annotate in EPIC-KITCHENS

Bounding Box degli Oggetti Attivi

I sostantivi presenti nelle narrazioni sono stati usati come riferimento per annotare gli oggetti con cui il partecipante interagiva (definiti "oggetti attivi") [3]. Per ogni sostantivo menzionato in un segmento di azione, sono stati estratti i fotogrammi all'interno e immediatamente prima/dopo il segmento temporale ($[t_s - 2s, t_e + 2s]$) [3]. Su questi frame, i lavoratori di AMT sono stati incaricati di disegnare i bounding box attorno agli oggetti specificati [3]. In totale, sono stati raccolti 454,255 bounding box [3].



Figura 4.3: Wordcloud degli oggetti interrogati di frequente

AMB

Partendo da queste annotazioni, è stato possibile creare la ground truth per *Active Memories Benchmark* (AMB). Essendo l'AMB un benchmark di tipo *visual-only QA*, uno dei principali problemi da affrontare è la selezione di una rappresentazione visiva degli oggetti per costruire le *visual query* (VQ). Per gestire l'occlusione tipica dei video egocentrici, causata dalle mani del soggetto o da altri oggetti, per ogni oggetto sono stati selezionati fino a tre *image patch* differenti, sufficientemente distanti temporalmente (almeno 0.5s tra una patch e l'altra) per mostrare pose differenti. Inoltre, sono state scelte patch con minima sovrapposizione spaziale con i bounding box di altri oggetti o mani attivi nello stesso frame.

Analogamente, per le location sono stati estratti i frame con minima sovrapposizione spaziale con gli oggetti attivi, in modo che la location fosse visibile senza la presenza di molti oggetti in movimento. Inoltre si è garantita una distanza minima temporale di 1s tra le immagini selezionate.

Sono stati selezionati casualmente 100 video di EPIC-KITCHENS, dai quali sono state generate circa 20.5K queries in maniera semi-automatica. Ogni domanda è composta da cinque possibili opzioni. In particolare, le possibili risposte vengono selezionate in maniera differente a seconda del tipo di domanda. Nel paper originale sono fornite linee guida generiche per i vari template, ad esempio:

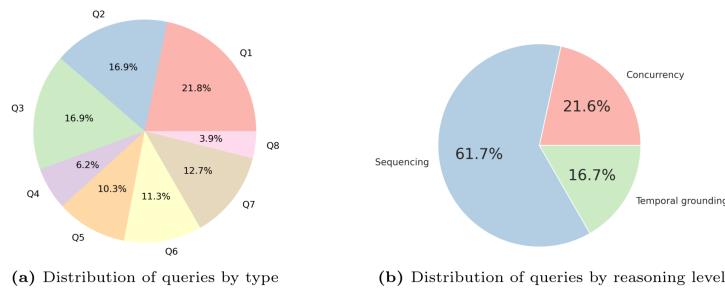


Figura 4.4: Distribuzioni dei tipi di query presenti in AMB

- **Q3-Q4:** le opzioni possono includere oggetti utilizzati con la mano opposta;
- **Q2-Q3:** il tempo della query t è fissato in modo tale che l'oggetto [VQ] non sia ancora stato utilizzato, richiedendo prima di individuare l'interazione con [VQ] e successivamente quelle precedenti o successive.
- **Q6:** le opzioni includono le location visitate immediatamente dopo che il soggetto ha interagito con un oggetto specifico;

4.2 ENIGMA-51

ENIGMA-51 è un dataset egocentrico acquisito in uno scenario industriale, in cui 19 soggetti hanno seguito istruzioni per completare operazioni di riparazione di schede elettroniche utilizzando strumenti industriali (ex: **cacciaviti elettrici**) e apparecchiature di laboratorio (ex: **oscilloscopi**).

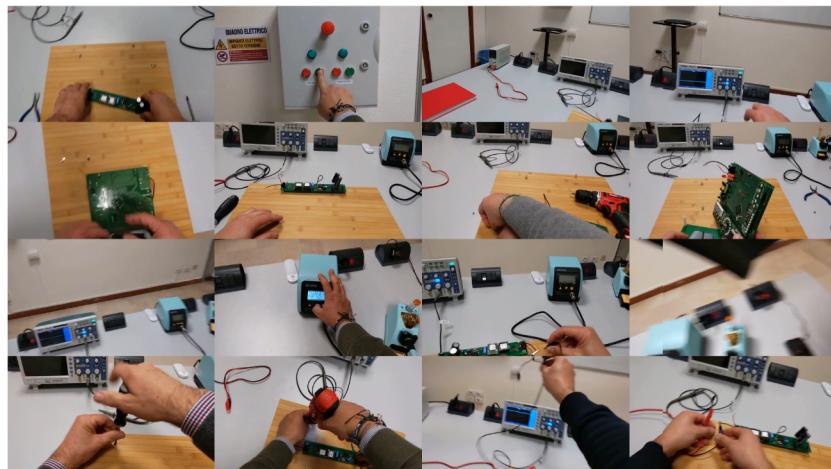


Figura 4.5: ENIGMA-51 dataset

Collezione dei dati

Le sequenze sono state acquisite in uno scenario controllato che riproduce fedelmente un ambiente industriale. I partecipanti hanno indossato un visore *Microsoft HoloLens 2*.

Le registrazioni sono caratterizzate da una risoluzione di 2272×1278 pixel e da un framerate di 30 fps. Ogni sequenza ha una durata media di circa 26 minuti, per un totale complessivo di 22 ore di materiale acquisito.

Durante le riprese, i partecipanti hanno seguito istruzioni dettagliate per completare due procedure di riparazione, una a bassa e una ad alta tensione. Le indicazioni venivano fornite direttamente attraverso l'HoloLens 2 tramite un'applicazione dedicata, che combinava audio, immagini e contenuti di realtà aumentata per guidare l'utente passo dopo passo.

Annotazioni

Le annotazioni coprono diversi livelli di dettaglio, dagli aspetti temporali alle interazioni mano-oggetto.

In tutti i 51 video sono stati individuati i *key frame* di interazione con oggetti. Ciascuno viene fornito di un timestamp e da un verbo che descrive l'azione in corso.

La tassonomia dei verbi comprende quattro azioni fondamentali:

- *first-contact* (primo contatto)
- *de-contact* (fine del contatto)
- *take* (prendere)
- *release* (rilasciare)

Nel complesso, sono state annotate **14.036 interazioni**.

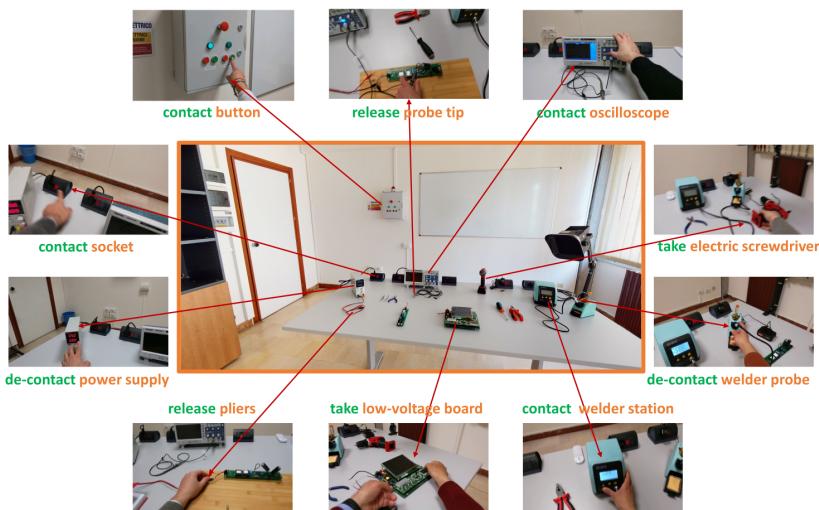


Figura 4.6: Annotazioni verbi in ENIGMA-51

Sono state definite 25 classi di oggetti, comprendenti sia oggetti fissi (ex: **pannello elettrico**) sia strumenti mobili (ex: **pinze**). Ogni oggetto è rappresentato attraverso una tupla che descrive la classe, le coordinate del rettangolo di delimitazione (*bounding box*) e lo stato, che specifica se l'oggetto è attivo o passivo nell'interazione. Complessivamente, sono stati annotati **275.135 oggetti**.

Sono stati etichettati i *bounding box* delle mani. Per ciascuna mano è stato corretto manualmente il lato (destra o sinistra) e indicata l'associazione con l'oggetto attivo in interazione. In totale, le mani annotate ammontano a **56.473**.

Insieme alle varie annotazioni, il dataset fornisce ulteriori risorse utili per lo studio delle interazioni in contesti industriali. Sono inclusi i **modelli**

3D dell’ambiente e degli oggetti, così come due **procedure** composte da istruzioni che guidano l’utente nelle interazioni con gli oggetti stessi.

Active Memories Benchmark (AMB)

La costruzione del benchmark ha usato metodologie analoghe a quelle riportate nel paper originale di AMEGO. I dettagli sono descritti nel Capitolo 5.

Di seguito sono riportate alcune statistiche rilevanti sul benchmark creato.

Nella figura Figure 4.7 si evidenziano le associazioni tra le categorie di oggetti presenti tra le domande e la risposta corretta.

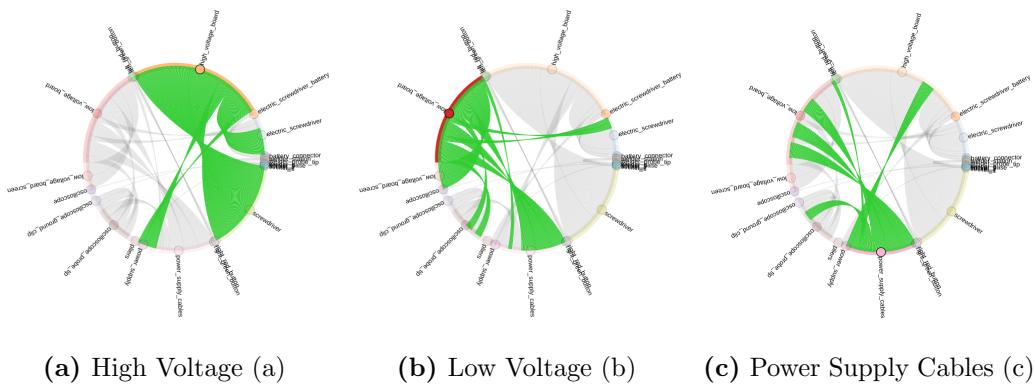


Figura 4.7: Chord diagrams che mostrano l’associazione tra categorie VQ e la risposta corretta

Si notano a primo impatto gli elementi centrali del dataset. In particolare, le schede elettroniche ad alta e bassa tensione risultano essere le più frequenti e vengono comunemente utilizzate insieme a: *cacciavite*, *sonda dell’oscilloscopio* e *cavi di alimentazione*. Per confermare queste osservazioni in maniera più quantitativa, è stata calcolata anche la matrice di correlazione tra le categorie (Fig. 4.8). Si conferma infatti che le interazioni più frequenti sono quelle con all’interno le schede elettroniche. Queste interagiscono più di frequente con *screwdriver*, *oscilloscope*, *power supply*. Questa caratteristica può essere notata anche nella Fig. 4.9.

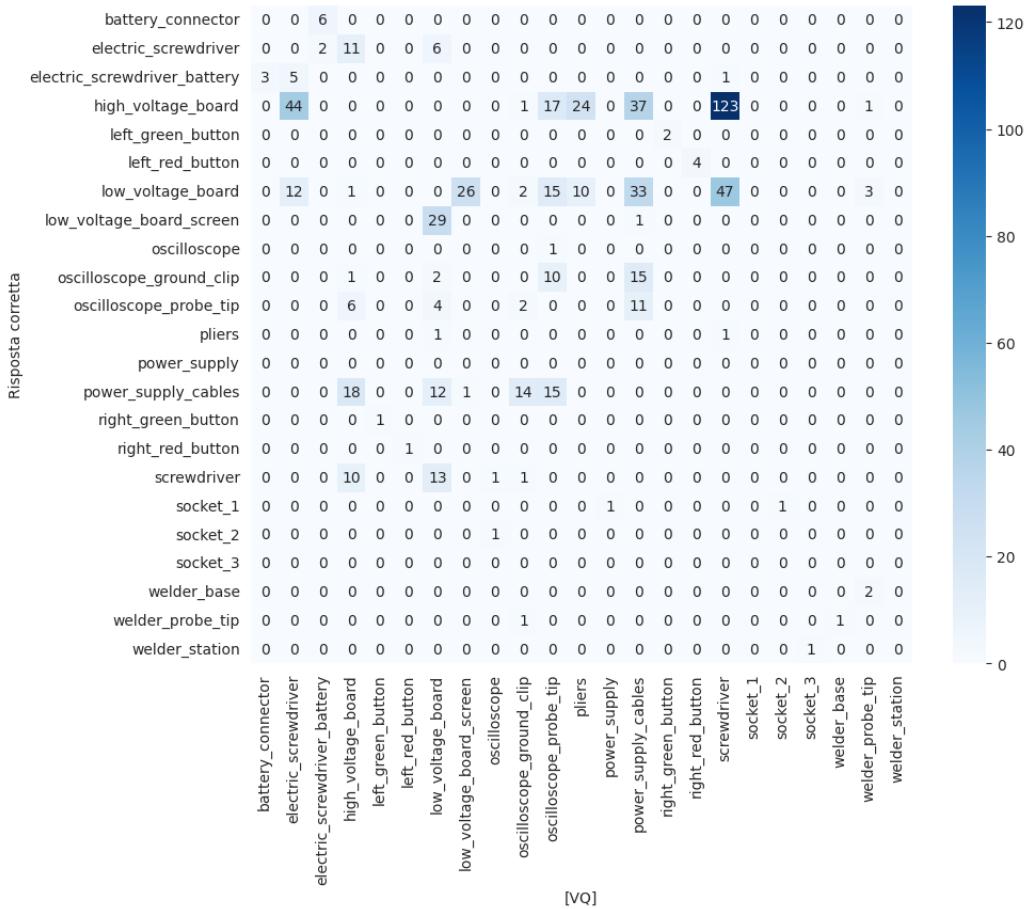


Figura 4.8: Heatmap della distribuzione delle categorie degli oggetti nel benchmark AMB per ENIGMA-51.

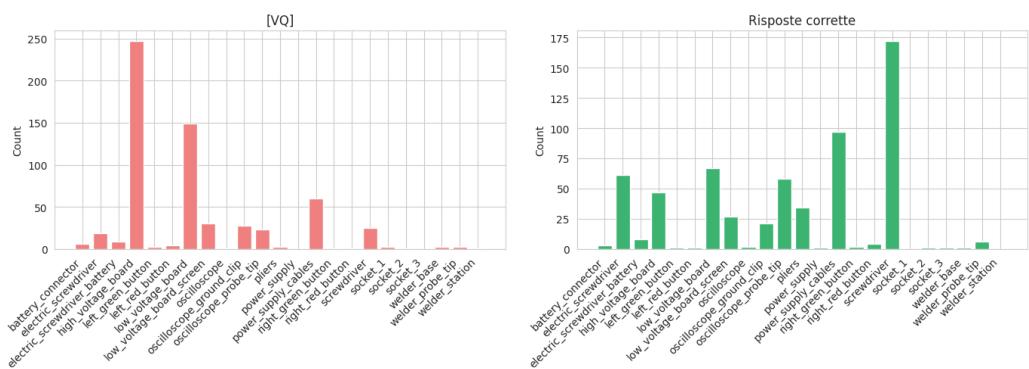


Figura 4.9: Istogramma delle occorrenze delle categorie degli oggetti nel benchmark AMB per ENIGMA-51.

Capitolo 5

Esperimenti

Gli esperimenti sono stati condotti seguendo i settaggi originali di AMEGO. Di seguito vengono riportate le principali impostazioni utilizzate.

Per l'identificazione delle interazioni mano-oggetto a livello di frame è stato utilizzato l'*Hand-Object Interaction detector* [32]. Le feature visive degli oggetti sono state estratte tramite il modello pre-addestrato *DINO-v2* [34], con ridimensionamento in fase di valutazione sulla versione *ViT-L*.

Il tracciamento degli oggetti durante le interazioni è stato gestito tramite l'EgoSTARK tracker [33], con i seguenti parametri:

- $\theta = 0.6$: soglia di similarità per l'associazione dei bounding box.
- $w_s = 30$: finestra temporale di frame utilizzata per garantire la coerenza spaziale.
- $s_o = 20$: numero minimo di bounding box consecutivi con forte sovrapposizione spaziale richiesto per definire un tracklet.
- $e_o = 20$: numero massimo di frame consecutivi senza rilevazioni associate prima di considerare un tracklet terminato, purché la mano rimanga visibile.

Come estrattore di feature per le location è stato utilizzato *SWAG* [35]. L'*optical flow* è stato stimato tramite il modello Flowformer [40], adottando i seguenti parametri:

- $\theta = 2000$: soglia applicata alla norma L2 dell'optical flow.
- $s_l = 5$: numero minimo di frame consecutivi richiesti per considerare attivo un *Location segment* l_j .

- $e_l = 5$: numero massimo di frame consecutivi per cui si tollera che la norma dell'optical flow superi la soglia massima oppure che non vengano rilevate mani; se tale limite viene superato, il segmento l_j viene terminato.
- $\tau = 0.5$: soglia minima di similarità richiesta per assegnare un segmento l_j a una location già esistente.

Di seguito vengono riportate le fasi principali di sperimentazione.

Il notebook e altre risorse che forniscono ulteriori dettagli sui lavori svolti sono disponibili nella repository GitHub del progetto:

<https://github.com/Kespers/Egocentric-Videos-Understanding-with-Active-Memory>

5.1 Setup ambiente

Per poter eseguire il progetto AMEGO in tutte le sue componenti, gli autori consigliano due ambienti `conda` separati, ciascuno con versioni di `python` e dipendenze differenti. In particolare:

- l'ambiente `amego`, basato su `python 3.9`, necessario per lanciare gli script nativi del framework;
- l'ambiente `handobj`, basato su `python 3.8`, per eseguire lo script di *hand-object detection*;

Il carico computazionale richiesto da questi task necessita dell'utilizzo di hardware apposito. Per questo motivo non è stato possibile condurre gli esperimenti su una macchina personale, ma si è fatto ricorso alle risorse messe a disposizione dal Dipartimento di Matematica e Informatica dell'Università di Catania.

In un contesto di questo tipo, è fondamentale garantire isolamento e riproducibilità degli ambienti di lavoro. Per questo motivo si è scelto di adottare un approccio basato su *Docker*, integrando i due ambienti `conda` in un unico container.

Il `Dockerfile` è stato strutturato partendo da Ubuntu come immagine di base. Su questa base è stato installato *Miniconda*, impiegato come gestore principale degli ambienti virtuali, permettendo così di mantenere all'interno dello stesso container due ambienti `conda` separati. È stata inoltre definita una directory di lavoro centrale, `/workspace/amego`, contenente il codice del progetto e gli script di setup. All'interno del container, nella directory `/workspace/ENIGMA-51`, è stata montata la partizione `/dataset/ENIGMA-51` del server, contenente l'intero contenuto del dataset, inclusi frame e annotazioni.

La build iniziale è stata tentata su una macchina equipaggiata con GPU **Tesla K80**. Tuttavia, la scheda video risulta troppo obsoleta e non supporta le versioni più recenti delle librerie necessarie.

Per l'esecuzione degli esperimenti è stato utilizzato un server con risorse più recenti, di seguito sono riportate le caratteristiche principali:

Componente	Specifiche principali
CPU	Intel(R) Xeon(R) Silver 4210 @ 2.20GHz
GPUs	3x Tesla V100S (32GB), 1x Tesla V100 (16GB)

Tabella 5.1: Specifiche hardware utilizzato.

Su questa infrastruttura, la build del container è andata a buon fine ed è stato possibile eseguire correttamente le procedure necessarie ad AMEGO.

In particolare, le operazioni principali sono state le seguenti:

1. **Download del video di test:** è stato scaricato il video P01_01 del dataset EPIC-KITCHENS [3] utilizzando gli script ufficiali forniti dai creatori del dataset [41].
2. **Estrazione dei frame:** il video è stato convertito in sequenze di frame.
3. **Hand-object detection:** generazione dei bounding box relativi alle hand-object-detections per ciascun frame.
4. **Estrazione Optical flow**
5. **AMEGO HOI / Location Segments:** creazione delle componenti di memoria di AMEGO
6. **Query:** esecuzione delle query AMB

Dopo aver verificato il corretto funzionamento dell'intera pipeline, è stato quindi possibile passare agli esperimenti principali su ENIGMA-51.

5.2 AMB su ENIGMA-51

Analogamente a quanto fatto per EPIC-KITCHENS, sono stati eseguiti gli script di AMEGO per estrarre le componenti indispensabili al funzionamento del metodo. La differenza è l'uso della partizione montata che ha agevolato la parte di estrazione frame in quanto erano già forniti.

Elaborazione files

Un primo lavoro ha riguardato la strutturazione delle cartelle. Data la struttura dei loader usati da AMEGO, è suggerita la seguente organizzazione:

```
<video_id>/
    rgb_frames/
        frame_0000000000.jpg
        frame_0000000001.jpg
        ...
    flowformer/
        flow_0000000000.pth
        flow_0000000001.pth
        ...
    hand-objects/
        <video_id>.pkl
```

In ENIGMA i frame dei video sono forniti in un'unica cartella con il formato:

<VIDEO-ID>_<FRAME-ID>.png

Inoltre la risoluzione fornita è (2272×1278), mentre AMEGO si aspetta (456×256). A tal proposito è stato sviluppato uno script Python che ristruttura e ridimensiona correttamente tutti i file.

Con queste basi è stato possibile creare i benchmark su cui testare la memoria. Come anticipato in precedenza, l'attenzione è stata posta sulle query di tipo **concurrency**, in particolare sulla query Q5. Inizialmente si era considerata anche la Q6, ma per motivi che saranno approfonditi in seguito, non è stato possibile includerla negli esperimenti.

Per la costruzione del benchmark su ENIGMA-51, le annotazioni disponibili sono state rielaborate al fine di generare i file JSON richiesti da *AMEGO* per l'esecuzione delle query.

Q5: What did I use with [VQ]?

La query Q5 valuta la capacità del modello di riconoscere oggetti utilizzati simultaneamente con un altro oggetto.

Il set di domande è strutturato come segue:

- **id:** Identificativo univoco della domanda
- **video_id:** Identificativo del video di riferimento

- **question:** Testo della domanda
- **question_image:** Crop del frame contenente l'oggetto VQ
- **answers:** Cinque possibili risposte
- **correct:** ID della risposta corretta

Creazione risposte

Per generare le query, si è proceduto selezionando le annotazioni in cui erano presenti almeno due interazioni *hand-object* simultanee tra oggetti di diversa categoria. Il primo oggetto dell'interazione è stato utilizzato come `question_image`, mentre le risposte sono state composte includendo:

- **RISPOSTA CORRETTA:** secondo oggetto dell'interazione;
- **RISPOSTE ERRATE:** vengono utilizzate le **classi** degli altri oggetti presenti nella scena come riferimento per selezionare lo stesso oggetto in altri punti dello *stesso* video (aspetto importante, poiché non è garantito che in video diversi compaiano gli stessi oggetti). Se gli oggetti generati nelle risposte sono meno di quattro, il set viene completato con classi scelte casualmente.

Tre versioni per ogni oggetto

Per ogni classe sono state selezionate tre patch differenti, temporalmente distinte di almeno 0,5 secondi l'una dall'altra, per mostrare diverse pose e ridurre gli effetti di occlusione tipici della visione egocentrica, ad esempio dovuti alle mani del soggetto o ad altri oggetti.

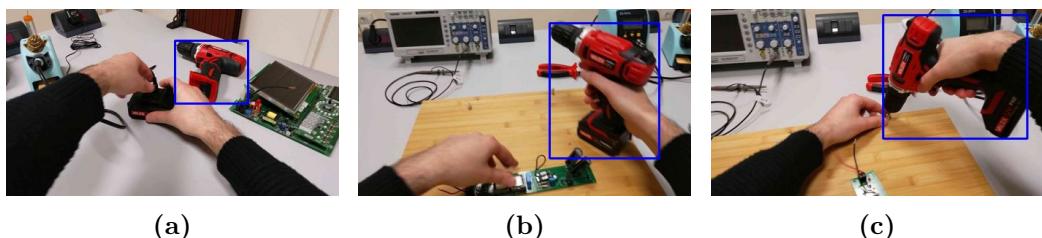


Figura 5.1: Versioni create per l'oggetto cacciavite

Conversione formato bounding box

Infine, è stata effettuata una conversione del formato dei bounding box: le annotazioni originali fornivano i box come

$$[x_{\min}, y_{\min}, \text{width}, \text{height}]$$

mentre AMEGO richiede:

$$[x_{\min}, y_{\min}, x_{\max}, y_{\max}]$$

Dopo questa operazione è stato quindi generato il file json per fare partire la query.

Esempio

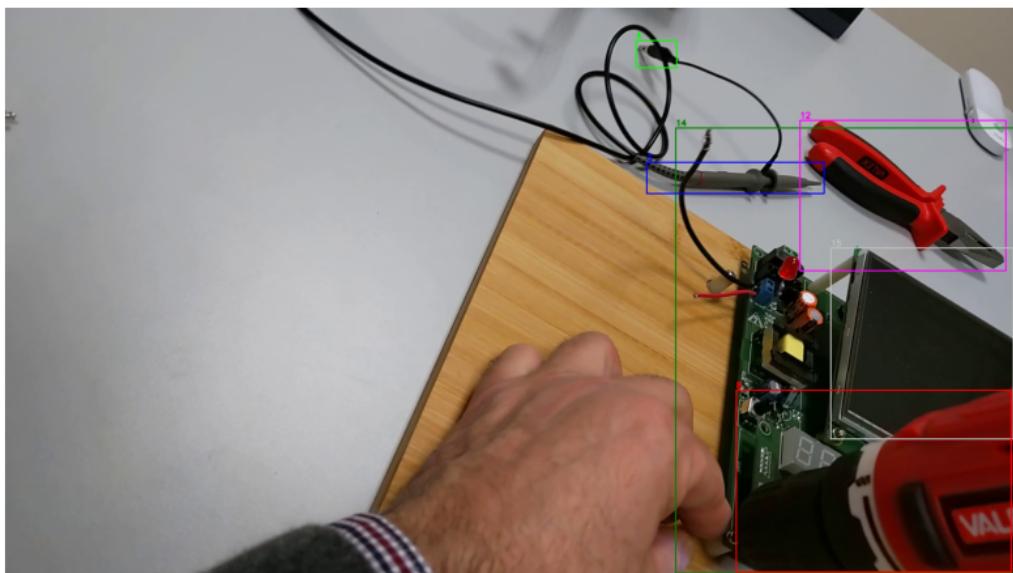


Figura 5.2: Frame 46_6043.jpg annotato con i seguenti oggetti e relativi indici/colori: 0: power_supply (grigio), 4: oscilloscope_ground_clip (verde), 12: pliers (magenta), 14: low_voltage_board (verde scuro), 3: oscilloscope_probe_tip (rosso), 15: low_voltage_board_screen (argento).

Come si nota dalla Fig. 5.2, l’interazione simultanea avviene tra **low voltage board** e **electric screwdriver**. Questi due elementi saranno quindi trattati rispettivamente come **VQ** e **CORRECT ANSWER**. In questo caso particolare, con gli altri oggetti presenti nella scena, riusciamo ad arrivare a 4 elementi.

Quindi, la query avrà la seguente forma:

```
{
  "id": "Q5_000000",
  "video_id": "46",
  "question": "What did I use with [VQ]?",
  "question_image": {
    "low_voltage_board": <INFO>
  },
  "answers": {
    "1": {"oscilloscope_ground_clip": <INFO>},
    "2": {"electric_screwdriver": <INFO>},
    "3": {"oscilloscope_probe_tip": <INFO>},
    "4": {"pliers": <INFO>},
    "5": {"low_voltage_board_screen": <INFO>}
  },
  "correct": 2
}
```

All'interno di ogni oggetto (<INFO>) troviamo le informazioni riguardo le 3 patch visuali. In particolare:

```
{
  "OBJECT": [
    [<frame_1>, [x0, y0, x1, y1]],
    [<frame_2>, [x0, y0, x1, y1]],
    [<frame_3>, [x0, y0, x1, y1]]
  ]
}
```

Q6: Where did I use [VQ]?

La query Q6 avrebbe testato la capacità del modello di identificare le location in cui è stato utilizzato un oggetto.

Tuttavia, le annotazioni di ENIGMA non fornivano informazioni sufficienti sulla posizione dei soggetti nei frame. In un primo momento si era ipotizzato di estrarre parole chiave dai file testuali delle procedure per inferire possibili location; tuttavia, il numero limitato di ambienti estraibili e la mancanza di corrispondenza con i frame annotati hanno reso questa sperimentazione poco praticabile.

In alternativa, sfruttando le location estratte da *AMEGO*, sono stati individuati **10** cluster.

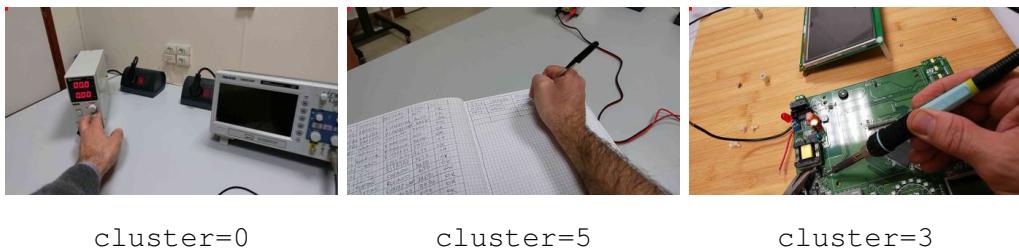


Figura 5.3: Esempi di location individuate da *AMEGO* con i relativi cluster assegnati.

Un'analisi manuale dei cluster ha mostrato che le possibili location corrispondono a: scrivania, area strumenti da tavolo, zona riparazione, consultazione, zona quaderno e quadro elettrico.

Sono però emerse alcune criticità: le location risultano spesso mescolate, con cluster che vengono considerati unici ma che in realtà dovrebbero essere suddivisi; inoltre, le riprese sono molto statiche e, per questa ragione, anche con annotazioni più accurate non si otterrebbero location realmente significative.

Alla luce di queste considerazioni, per questo tipo di query sarebbe più opportuno utilizzare un dataset industriale differente, caratterizzato da una maggiore varietà di postazioni, da cui costruire un benchmark più affidabile. Per tali motivi, il lavoro si è concentrato esclusivamente sulla query Q5.

5.3 Analisi

Tempi di esecuzione

Di seguito sono riportati i tempi approssimativi di esecuzione dei vari script, eseguiti in sequenza sul test-set ENIGMA-51. Si nota come la fase più dispendiosa sia l'estrazione delle hand-object detections e dell'optical flow, che richiedono circa 30 ore di calcolo.

Tabella 5.2: Tempi di elaborazione dei vari script

Script	Tempo
HOI	$\approx 30\text{h}$
OPTICAL FLOW	$\approx 30\text{h}$
AMEGO HOI TRACKLETS	$\approx 26\text{h}$
LOCATION SEGMENTS	$\approx 8\text{h}$
Q5 QUERY	$\approx 30\text{min}$

Visualizzazione cluster

Come anticipato nella discussione relativa alla query Q6, è stato sviluppato uno script che consente di visualizzare in modo approfondito la memoria generata. Lo script crea due cartelle principali: **HOI FRAMES** e **LS FRAMES**. All'interno di ciascuna cartella vengono generate sottocartelle corrispondenti ai cluster individuati, contenenti i relativi frame. Nel caso di HOI_FRAME, per ogni immagine viene inoltre disegnato il bounding box corrispondente.

```

HOI_FRAMES/
<cluster_id>/
  <track_id>
    frame.jpg
    frame.jpg
    frame.jpg
  <track_id>
  <track_id>

```

```

LS_FRAMES/
<cluster_id>/
  frame.jpg
  frame.jpg
  frame.jpg

```

Visualizzazione benchmark

Per avere sempre a disposizione le diverse query, è stato sviluppato uno script che organizza le domande nella seguente struttura:

```

<video_id>/
  Q5_00000x/
    VQ/
      <image_patch_1>.jpg
      <image_patch_2>.jpg
      <image_patch_3>.jpg
    Answers/
      <answer_ID>_<class_name>/
        <image_patch_1>.jpg
        <image_patch_2>.jpg
        <image_patch_3>.jpg

```

Visualizzazione video

Per avere una visione d'insieme della memoria creata da AMEGO, è stato realizzato uno script che, dato un video, raggruppa i vari HOI tracklets trovati

e li sovrappone con i rispettivi bounding. Come mostrato in Fig. 5.4, nella parte superiore del video sono visualizzate informazioni generali sul tipo di video e sul frame corrente. Al centro scorrono i frame con disegnati i bounding box dei vari HOI, ciascuno caratterizzato da un colore assegnato univocamente a ogni cluster.

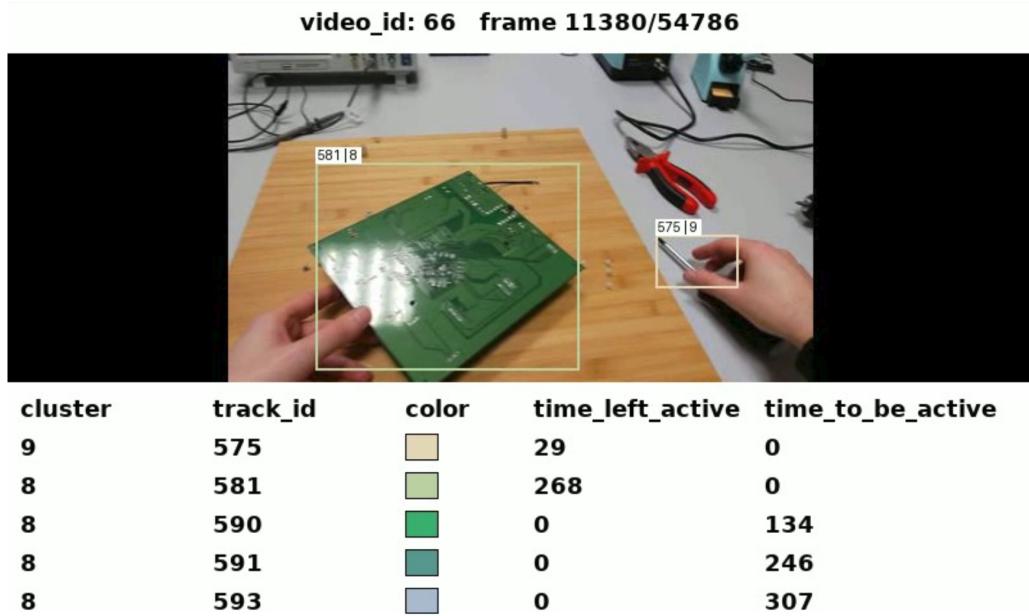


Figura 5.4: visualizzazione degli HOI tracklet del video 66

Al di sotto del video, una tabella mostra alcune informazioni chiave sugli HOI tracklets visualizzati. In particolare, si riporta il **Track ID**, il **Cluster** associato con indicazione dell'ID e del colore univoco, il numero di frame rimanenti in cui il tracklet è ancora attivo (*Time Left Active*) e il numero di frame prima che il tracklet diventi attivo (*Time to be Active*).

Capitolo 6

Risultati

In questo capitolo vengono presentati i risultati ottenuti. Si inizia con una panoramica dei risultati riportati nel lavoro originale, al fine di avere un riferimento di partenza, per poi illustrare i risultati sperimentali ottenuti sul nuovo dataset.

6.1 EPIC-KITCHENS

In principio AMEGO è stato valutato sul dataset EPIC-KITCHENS, confrontando le prestazioni con diverse baseline comunemente adottate nel task di video-QA:

- **Semantic-free QA (SF-QA)** utilizza modelli vision-language, come CLIP [42], per mappare query, video e risposte nello stesso spazio di embedding. Le feature visive vengono estratte dai frame del video, dai patch delle query e dalle risposte, mentre le feature testuali provengono dalla domanda. L'embedding della query è ottenuto come media delle feature, e la risposta con la similarità più alta viene selezionata.
- **SF-QA (obj)** è una variante di SF-QA che include anche le feature visive degli oggetti attivi rilevati da [32].
- **Semantic QA (S-QA)** sfrutta *captioner* pre-addestrati per generare un sommario semantico del video. Si usano LaViLa [43] per il video egocentrico e BLIP-2 [44] per i patch della query. Le caption vengono poi passate a LLaMA-2-7B [45] per rispondere alle domande. Se il testo supera i 4096 token, viene sottocampionato.

- **Multi-round semantic QA (LLoVi)** [46] funziona in due round: prima sintetizza le caption del video alla luce della domanda, poi risponde alla query usando il sommario generato.

I risultati di AMEGO si dividono in: AMEGO-S e AMEGO-L, a seconda della dimensione del visual feature extractor (ViT-S/B vs ViT-L).

Tabella 6.1: Accuracy (%) sulle diverse query di AMB. Migliori valori in grassetto.

Method	SQ				CO		TG		Total
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	
Random	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
SF-QA	13.7	21.6	22.5	26.8	22.1	31.9	23.7	26.2	22.0
SF-QA (obj)	13.1	23.4	22.6	23.2	21.7	26.1	23.8	25.2	21.2
S-QA (LaViLa)	20.9	20.6	21.2	24.6	24.9	27.1	21.4	22.6	22.4
S-QA (BLIP-2)	23.9	22.0	22.5	23.3	27.5	27.0	20.2	24.1	23.6
S-QA (LaViLa+BLIP-2)	22.8	22.2	21.4	22.6	25.1	26.1	21.4	24.5	22.9
LLoVi (LaViLa)	21.1	20.2	20.8	21.0	21.2	20.3	20.5	21.6	20.8
LLoVi (BLIP-2)	22.3	21.4	21.8	22.2	25.6	26.7	18.1	22.2	22.4
LLoVi (LaViLa+BLIP-2)	22.8	21.9	21.5	24.6	25.3	26.5	18.5	19.8	22.6
AMEGO - S	32.0	35.1	34.8	35.8	24.7	37.8	33.6	44.3	33.8
AMEGO - L	33.7	36.3	37.2	38.3	27.6	44.3	34.7	48.9	36.3

Tutte le baseline ottengono risultati migliori sulle query relative alla *currency*, probabilmente perché sfruttano pattern ricorrenti presenti nei dati di addestramento, ad esempio una padella spesso utilizzata sul piano cottura [1]. Tuttavia, le performance complessive rimangono vicine alla soglia della scelta casuale.

AMEGO invece ottiene buoni risultati su tutte le tipologie di query, superando le baseline con un margine consistente (+12.7%). La domanda in cui AMEGO mostra maggior difficoltà è Q5, a causa dei limiti attuali dei detector di interazione mano-oggetto nel predire oggetti multipli che interagiscono contemporaneamente con la stessa mano del soggetto.

6.2 ENIGMA-51

Le valutazioni sono state effettuate sul test-set di ENIGMA-51, che comprende i seguenti video:

$$\{46, 47, 49, 53, 65, 66, 85, 86, 88, 89, 95, 107, 131, 141, 143, 144\}$$

Un aspetto rilevante è che, in presenza di più oggetti con lo stesso punteggio, AMEGO seleziona quello corretto in modo casuale. Tale meccanismo

introduce una variabilità nei risultati; per stimarne in maniera più robusta le prestazioni, l'esperimento è stato ripetuto per circa 100 iterazioni, calcolando i valori medi di accuracy. In media, il modello ha raggiunto una **accuracy totale del 21.94%**. Nella tabella 6.2 viene riportata l'accuracy raggruppata per ogni video.

Tabella 6.2: Accuracy media (%) sul test-set di enigma. Migliori valori in grassetto.

Video ID	Random (%)	AMEGO - L (%)
46	20	18.23
47	20.0	27.36
49	20.0	35.76
53	20	10.87
65	20.0	26.86
66	20	9.19
85	20.0	37.37
86	20.0	26.56
88	20.0	24.32
89	20.0	33.90
95	20.0	21.60
107	20	15.46
131	20.0	35.25
141	20	18.21
143	20	7.71
144	20.0	30.81
Total	20.0	23.72

Si osservano risultati non sempre soddisfacenti: in alcuni video l'accuracy scende al di sotto della soglia casuale (20%), mentre in altri supera sensibilmente le prestazioni originali. Analizzando le percentuali per ogni domanda si evidenzia che **203** query hanno totalizzato accuracy nulla. Per approfondire la natura dei risultati, si analizza la query Q5_000089, scelta come caso esemplificativo per evidenziare la criticità principale riscontrata.

Come si nota nella Fig. 6.3, nonostante le patch della VQ e delle varie risposte appaiono ben distinte, il modello non seleziona mai la risposta corretta. Per comprendere tale comportamento è utile osservare la struttura della memoria di AMEGO, fornita come file JSON. Di particolare importanza sono i seguenti campi:

- `track_id`: identificativo univoco del tracklet.
- `obj_bbox`: bounding box dell'oggetto
- `num_frame`: lista dei frame in cui l'oggetto è rilevato durante l'interazione.

- **cluster**: ID d’istanza assegnato al tracklet per raggruppare interazioni simili dello stesso oggetto.

Come discusso nei capitoli precedenti, AMEGO risponde alle query valutando la presenza di *overlap* temporali tra oggetti che il modello considera appartenenti allo stesso identificativo (**cluster**). Il problema può quindi risiedere nell’assegnazione delle istanze ai cluster.

Per approfondire l’analisi, è stato utilizzato lo script di visualizzazione dei cluster discusso nel Capitolo 5. Dall’analisi emerge che il modello di hand-object detection riesce a individuare correttamente gli oggetti a livello generale, e il *tracker* mantiene buone coerenze spaziali. Tuttavia, sorge un limite significativo legato al livello di dettaglio richiesto: nelle annotazioni di ENIGMA vengono specificati particolari molto piccoli come connettori o il tipo di scheda. Come mostrato in Fig. 6.1, il modello, non essendo stato addestrato su dataset industriali, tende a raggruppare in modo grossolano elementi complessi, arrivando a considerare l’intero quadro elettrico come un’unica istanza.

Un ulteriore limite emerge quando oggetti visivamente simili vengono confusi tra loro. Come mostrato in Fig. 6.2, bastano piccole somiglianze, anche solo nel colore o nella forma, affinché il modello li raggruppi nello stesso cluster. Questo porta a difficoltà nel distinguere strumenti affini, come diversi tipi di cacciaviti, pinze o saldatori, oppure componenti che ricordano visivamente un cavo.

Questo comportamento ricorre con una certa frequenza nei cluster. Per valutare se tale fenomeno potesse effettivamente influenzare i risultati, è stata adottata una strategia di “rilassamento” delle regole di valutazione dell’accuracy, considerando come corrette anche risposte associate a oggetti visivamente molto simili. In particolare, sono state applicate le seguenti indicazioni:

- **Socket**: nelle annotazioni sono presenti quattro tipologie distinte, qui considerate come un’unica categoria.
- **Bottoni**: unificazione di tutte le varianti.
- **Board**: accorpamento delle diverse tipologie di schede.
- **Cavi**: unificazione tra power supply, probe tips e clips.

Applicando questo rilassamento, l’accuracy raggiunge il valore di **28.86%**, molto vicino a quello riportato nel paper originale.



Figura 6.1: HOI relativo al cluster=2 con track_id = 10.

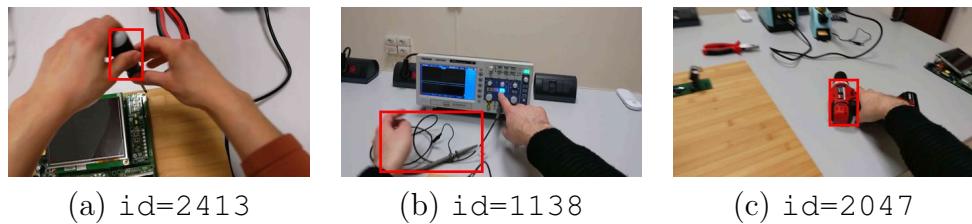


Figura 6.2: HOI relativi al cluster=24

Tabella 6.3: VQ e risposte per la query Q5_000089: per ogni classe sono mostrate tre versioni.

Tipo	Versione 1	Versione 2	Versione 3
VQ			
ANS_1			
ANS_2			
ANS_3			
ANS_4			
ANS_5			

Esempi

Query con risposta corretta: Q5_000388

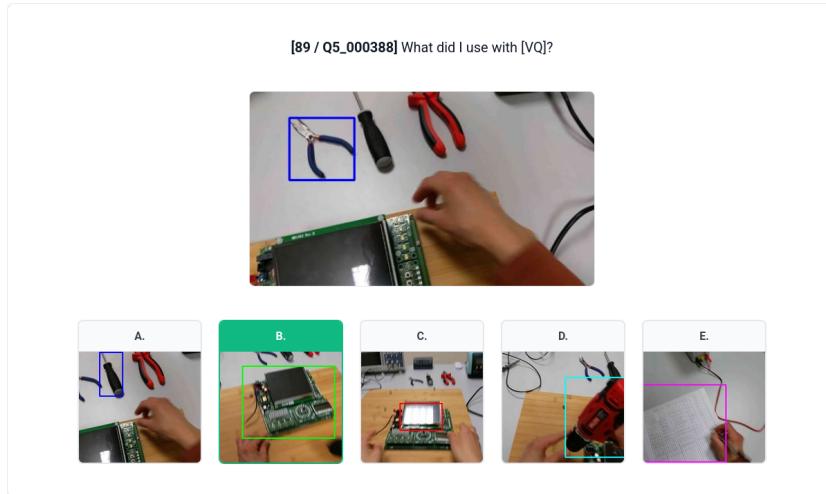


Figura 6.3: ENIGMA-51_AMB: Q5_000388. In verde la risposta corretta

Analizzando il filmato prodotto con lo script discusso nel Cap. 5, notiamo come AMEGO abbia riconosciuto correttamente l'overlap dei due oggetti ed abbia probabilmente usato questa informazione per rispondere a questa query.

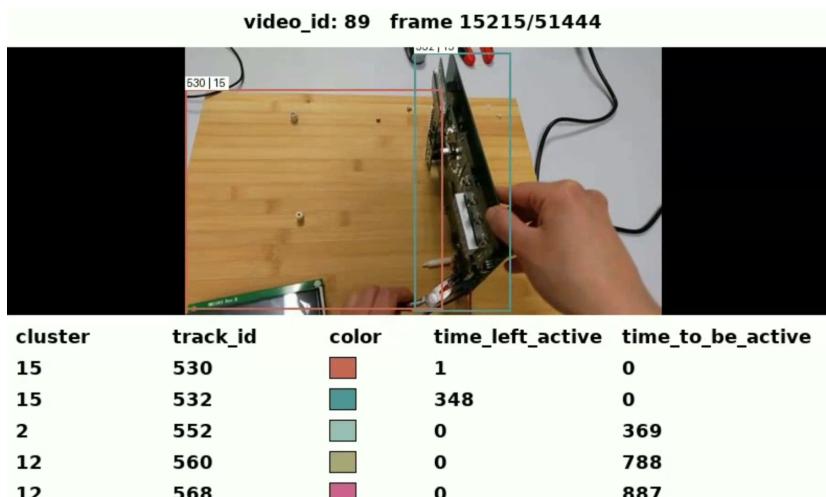


Figura 6.4: ENIGMA-51: Video 89 con annotazioni prodotte dalla memoria AMEGO

Query con risposta errata: Q5_000104

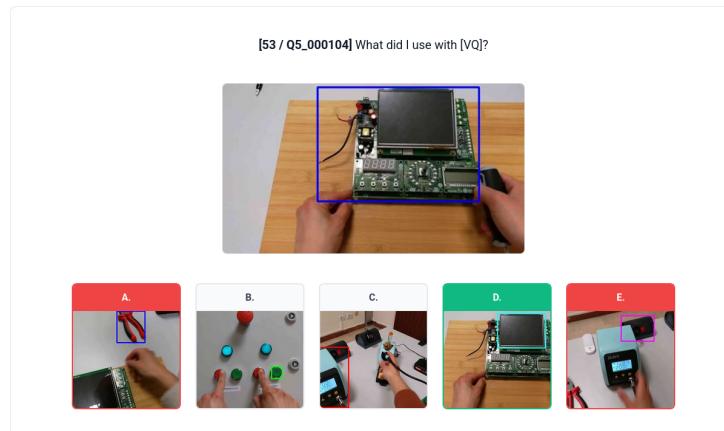


Figura 6.5: ENIGMA-51_AMB: Q5_000104. In verde la risposta corretta, in rosso quelle selezionate erroneamente dal modello.

Si può notare innanzitutto la difficoltà nell'individuare la risposta corretta: la *board* viene percepita come un unico oggetto, mentre in questo caso si distingue tra l'interazione con lo schermo della board e quella con la board stessa.

In particolare, durante l'inferenza ha considerato equivalenti le risposte **A** ed **E**, distribuendo le predizioni in maniera quasi equa tra le due. Questo comportamento è probabilmente dovuto alla forte somiglianza visiva tra i dettagli delle due opzioni.

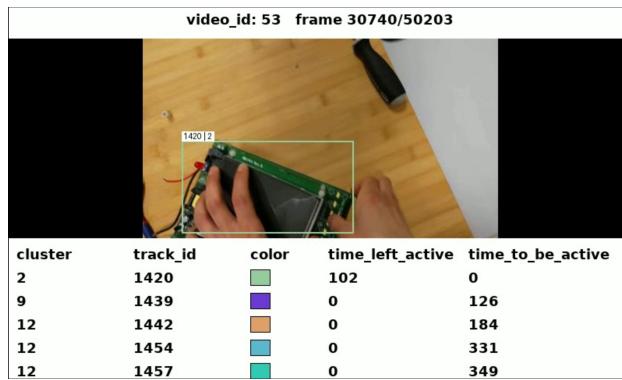


Figura 6.6: ENIGMA-51: Video 53 con annotazioni prodotte dalla memoria AMEGO

Come previsto, analizzando il video estratto dalla memoria (Fig. 6.6), si nota che il modello ha percepito la *board* come un unico oggetto.

Conclusione

In questa tesi abbiamo analizzato l'importanza dei video egocentrici come strumento per comprendere e modellare il comportamento umano in diversi contesti.

A tal fine, è stato introdotto AMEGO [1], un framework *semantic-free* per la rappresentazione di Human-Object Interaction in video egocentrici, che consente di generare una memoria attiva di oggetti e interazioni senza fare affidamento su etichette semantiche predefinite. Sono stati descritti i componenti principali del modello e il benchmark creato ad hoc, con particolare attenzione al task di *query sequencing*.

Per valutare queste capacità semantic-free, è stato utilizzato il dataset industriale ENIGMA-51 [2], opportunamente adattato per costruire un insieme di test coerente con il benchmark di AMEGO. Si è proceduto alla creazione del benchmark esclusivamente per Q5, rispettando i vincoli temporali tra le patch relative alle diverse classi e apportando una serie di affinamenti per rendere il dataset più significativo per la valutazione.

I risultati sperimentali hanno evidenziato alcuni limiti del modello, soprattutto in contesti industriali, dove oggetti visivamente simili ma semanticamente differenti (ad esempio diversi tipi di cacciaviti o componenti elettrici) vengono talvolta considerati come appartenenti a un unico gruppo. Questo evidenzia la necessità di strategie più robuste per il raggruppamento degli oggetti, poiché la logica di AMEGO, in particolare per la query analizzata, si basa in gran parte sulla corretta assegnazione delle classi. La presenza di errori in questa fase ha generato un livello significativo di rumore, che si riflette direttamente nei risultati osservati.

Un possibile miglioramento consiste nel fine-tuning dei modelli di estrazione delle feature e di hand-object detection su dataset industriali, in modo da aumentare la capacità di distinguere oggetti visivamente simili e di cogliere dettagli a diversi livelli di precisione. Su un dataset industriale, tale approccio potrebbe migliorare significativamente la rilevazione e la classificazione di elementi specifici tipici di questi contesti. Tuttavia, questa modifica potrebbe ridurre la natura semantic-free del modello, favorendo

l'apprendimento di classi strettamente legate al dominio industriale.

Un ulteriore miglioramento potrebbe riguardare l'uso di frame ad alta risoluzione. L'impiego delle risoluzioni originali dei dataset, sebbene più oneroso dal punto di vista computazionale, potrebbe favorire una migliore rilevazione dei dettagli critici degli oggetti.

Nonostante questi limiti, *AMEGO* costituisce una solida base per lo studio dei video egocentrici. Il principale vincolo risiede nella forte dipendenza dai modelli di supporto, che mostrano difficoltà nel generalizzare in modo completamente indipendente dalle classi specifiche. Tale limitazione emerge in maniera più evidente nella Q5, la quale si basa quasi interamente su questi strumenti, risultando nei punteggi più bassi anche nel paper originale. Questo scenario evidenzia però il margine di miglioramento più ampio.

Per sviluppi futuri, sarebbe quindi interessante esplorare innanzitutto altre tipologie di query, considerando anche l'eventuale utilizzo di dataset alternativi. Sarebbe inoltre utile valutare il comportamento del modello utilizzando strumenti class-agnostic differenti, poiché costituiscono la struttura portante del modello e potrebbero influenzarne significativamente le prestazioni.

Bibliografia

- [1] Gabriele Goletto, Tushar Nagarajan, Giuseppe Averta, and Dima Damen. Amego: Active memory from long egocentric videos. In *European Conference on Computer Vision*, 2024.
- [2] Francesco Ragusa, Rosario Leonardi, Michele Mazzamuto, Claudia Bonanno, Rosario Scavo, Antonino Furnari, and Giovanni Maria Farinella. Enigma-51: Towards a fine-grained understanding of human-object interactions in industrial scenarios. *IEEE Winter Conference on Application of Computer Vision (WACV)*, 2024.
- [3] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 43(11):4125–4141, 2021.
- [4] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding, 2023.
- [5] Xitong Yang, Fu-Jen Chu, Matt Feiszli, Raghav Goyal, Lorenzo Torresani, and Du Tran. Relational space-time query in long-form videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6398–6408, June 2023.
- [6] Ziyu Ma, Chenhui Gou, Hengcan Shi, Bin Sun, Shutao Li, Hamid Rezatofighi, and Jianfei Cai. Drvideo: Document retrieval based long video understanding, 2024.
- [7] Jongwoo Park, Kanchana Ranasinghe, Kumara Kahatapitiya, Wonjeong Ryu, Donghyun Kim, and Michael S. Ryoo. Too many frames, not all useful: Efficient strategies for long-form video qa, 2025.

- [8] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent, 2024.
- [9] Ying Wang, Yanlai Yang, and Mengye Ren. Lifelongmemory: Leveraging llms for answering queries in long-form egocentric videos, 2024.
- [10] Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos, 2025.
- [11] Chao-Yuan Wu, Yanghao Li, Karttikayaa Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition, 2022.
- [12] Yunxin Li, Xinyu Chen, Baotain Hu, and Min Zhang. Llms meet long video: Advancing long video question answering with an interactive visual adapter in llms, 2024.
- [13] Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models, 2024.
- [14] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding, 2024.
- [15] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, Yan Lu, Jenq-Neng Hwang, and Gaoang Wang. Moviechat: From dense token to sparse memory for long video understanding, 2024.
- [16] Anurag Arnab, Chen Sun, and Cordelia Schmid. Unified graph structured models for video understanding, 2021.
- [17] Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos, 2018.
- [18] Yuren Cong, Wentong Liao, Hanno Ackermann, Bodo Rosenhahn, and Michael Ying Yang. Spatial-temporal transformer for dynamic scene graph generation, 2021.

- [19] Ashesh Jain, Amir R. Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs, 2016.
- [20] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as composition of spatio-temporal scene graphs, 2019.
- [21] Chih-Yao Ma, Asim Kadav, Iain Melvin, Zsolt Kira, Ghassan AlRegib, and Hans Peter Graf. Attend and interact: Higher-order object interactions for video understanding, 2018.
- [22] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network, 2018.
- [23] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs, 2018.
- [24] Will Price, Carl Vondrick, and Dima Damen. Unweavenet: Unweaving activity stories, 2022.
- [25] Ivan Rodin, Antonino Furnari, Kyle Min, Subarna Tripathi, and Giovanni Maria Farinella. Action scene graphs for long-form understanding of egocentric videos, 2023.
- [26] Yong Jae Lee, Joydeep Ghosh, and Kristen Grauman. Discovering important people and objects for egocentric video summarization. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1346–1353, 2012.
- [27] Zheng Lu and Kristen Grauman. Story-driven summarization for egocentric video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.
- [28] Bo Xiong and Kristen Grauman. Detecting snap points in egocentric video with a web photo prior. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 282–298, Cham, 2014. Springer International Publishing.
- [29] Yen-Liang Lin, Vlad I. Morariu, and Winston Hsu. Summarizing while recording: Context-based highlight detection for egocentric videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2015.
- [30] Bin Zhao and Eric P. Xing. Quasi real-time summarization for consumer videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

- [31] Bo Xiong, Gunhee Kim, and Leonid Sigal. Storyline representation of egocentric videos with an applications to story-based search. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.
- [32] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F. Fouhey. Understanding human hands in contact at internet scale, 2020.
- [33] Hao Tang, Kevin Liang, Matt Feiszli, and Weiyao Wang. Egotracks: A long-term egocentric visual object tracking dataset, 2023.
- [34] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024.
- [35] Mannat Singh, Laura Gustafson, Aaron Adcock, Vinicius de Freitas Reis, Bugra Gedik, Raj Prateek Kosaraju, Dhruv Mahajan, Ross Girshick, Piotr Dollár, and Laurens van der Maaten. Revisiting weakly supervised pre-training of visual perception models, 2022.
- [36] Amazon Mechanical Turk. Amazon mechanical turk, 2005. Accessed: 2025-08-25.
- [37] Google. Google cloud speech api. <https://cloud.google.com/speech>. Accessed: 2025-08-25.
- [38] IBM. Ibm watson speech to text. <https://www.ibm.com/watson/services/speech-to-text>. Accessed: 2025-08-25.
- [39] Carnegie Mellon University. Cmu sphinx. <https://cmusphinx.github.io/>. Accessed: 2025-08-25.
- [40] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer: A transformer architecture for optical flow, 2022.
- [41] EPIC-KITCHENS. Epic-kitchens download scripts. <https://github.com/epic-kitchens/epic-kitchens-download-scripts>. Accessed: 2025-08-26.

- [42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.
- [43] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models, 2022.
- [44] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.
- [45] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.
- [46] Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. A simple llm framework for long-range video question-answering, 2024.