Socioeconomic Classification in Metropolitan Lima Using Google Street View Images and Machine Learning

*Link del código: https://github.com/m41k1204/ml_NSE_classifier

Michael Hinojosa
Faculty of Computing
Computer Science
michael.hinojosa@utec.edu.pe
Contribución: 100%

Jorge Quenta
Faculty of Computing
Data Science
jorge.quenta@utec.edu.pe
Contribución: 100%

Mikel Bracamonte
Faculty of Computing
Computer Science
mikel.bracamonte@utec.edu.pe
Contribución: 100%

Eduardo Aragon
Faculty of Computing
Computer Science
eduardo.aragon@utec.edu.pe
Contribución: 100%

***Abstract*—This study presents a machine learning approach for classifying socioeconomic levels in the Metropolitan Area of Lima using Google Street View images. We developed a dataset of 15,000 images from 12 districts representing three socioeconomic categories (High, Medium, Low) based on official INEI data. Visual features were extracted using DINOv2, a self-supervised vision transformer, producing 1536-dimensional embeddings. Four models were evaluated: Logistic Regression, SVM, MLP, and XGBoost. SVM with RBF kernel achieved the highest F1-Score of 91.07%, followed by MLP (89.95%) and Logistic Regression (88.67%). Dimensionality reduction analysis revealed that socioeconomic classes form a continuous nonlinear manifold rather than distinct clusters, explaining the superior performance of nonlinear models. Results demonstrate that street-level images combined with pre-trained vision transformers effectively identify socioeconomic patterns, offering a scalable alternative to traditional census methods for urban monitoring.**

I. Introduction

Socioeconomic indicators are essential tools for designing and monitoring the impact of public policies on urban populations [1]. Understanding the spatial distribution of socioeconomic level in metropolitan areas enables local authorities and policymakers to identify vulnerable populations and better allocate much needed resources to those who need them the most.

Traditional methods for obtaining socioeconomic data rely on national censuses conducted every 10 years, which can be helpful with populations that do not change much. However, in rapidly evolving cities like the Metropolitan Area of Lima (MAL), these methods present critical limitations, as constant demographic growth, internal and external waves of immigration, and economic transformations demand more frequent monitoring capabilities.

Over the years, Google Street View (GSV) and Google Maps have emerged as the go to services for seeing any urban areas worldwide, being extremely useful for tourists, but also for studies that aim to use street or satelite images in socioeconomic studies [1] [2]. The *MAL*, with over 10 million inhabitants, is one of the largest metropolitan areas in South America. The city has grown rapidly due to several waves of internal immigration and therefore creating significant socioeconomic differences, with informal settlements, Medium-class neighborhoods, and wealthy districts often located close to each other.

This work addresses these gaps by implementing multiple machine learning models to predict the Socioeconomic Status (Alto, Medio, Bajo) of neighborhoods in the Metropolitan Area of Lima based on Google Street View images. We construct a dataset of 15,000 street-view images and implement four classification approaches: Logistic Regression as a baseline model, Support Vector Machines (SVM), Multilayer Perceptron (MLP) and XGBoost. Visual features are extracted using a pre-trained DINOv2 model from Meta AI, and model performance is evaluated using appropriate metrics for multi-class classification.

II. Related Work

A. Urban Classification using Satellite images

Rahman et al. [2] classified urban areas in developing countries using satellite images and deep learning. They studied several cities in fast-growing countries including Lima. Their approach used Google Earth images covering 50×50 km areas for each city.

The authors proposed a categorization method that divides urban environments into 16 subcategories based on two dimensions: the state of urbanization and the architectural form of buildings. These subcategories were then grouped into four socioeconomic classes: highly informal, moderately informal, moderately formal, and highly formal. They trained three deep learning models (FCN-8, U-Net, and DeepLabv3+) to automatically segment satellite images into these four categories.

For the Metropolitan Area of Lima, their DeepLabv3+ model achieved 96.75% accuracy, which was the highest among all seven cities studied. This work showed that images can be used to classify socioeconomic status.

B. Socioeconomic Prediction from Street-View images

Machicao et al. [1] took a different approach by using Google Street View (GSV) images instead of satellite images. They focused on Vale do Ribeira in Brazil, a semi-rural region. Their study aimed to predict income-based socioeconomic indicators using street-level images.
The authors collected images from 28,092 locations. They used a pre-trained VGG-16 network as a feature extractor.
Their model predicted income levels across five classes, achieving 55% exact accuracy and 80% accuracy with 1 class tolerance. The best predictions were for the highest income class (80% correct), while lower income areas were harder to classify.
The study showed that street-view images can be used to estimate socioeconomic indicators, although it was conducted in a semi-rural environment.

Both studies demonstrate that machine learning combined with publicly available images can help identify socioeconomic levels.
However, Rahman et al. focused on satellite images for urban classification in the *MAL* amongst other cities and Machicao et al. used street-view images for income prediction in a semi-rural area.

III. Objectives

A. Principal Objective

This project seeks to study the classification of socioeconomic levels in the Metropolitan Area of Lima (MAL), a highly urbanized area, by training machine learning models with 15,000 images from different districts of the city.

B. Specific Objectives

- Create a dataset of 15,000 images from the *MAL* and label them by category
- Extract feature vectors from the 15,000 images using DINOv2
- Train four machine learning models with the dataset
- Evaluate the models using metrics such as Accuracy, Precision, Recall, and F1-Score

IV. Methodology

This project is based on a study published in 2020 by the National Institute of Statistics and Informatics of Peru (INEI) [3]. The study presents the socioeconomic levels of all districts in the *MAL* at the city block level.

Figure 1 shows an example of how city blocks are categorized by socioeconomic levels in the *La Victoria* district.

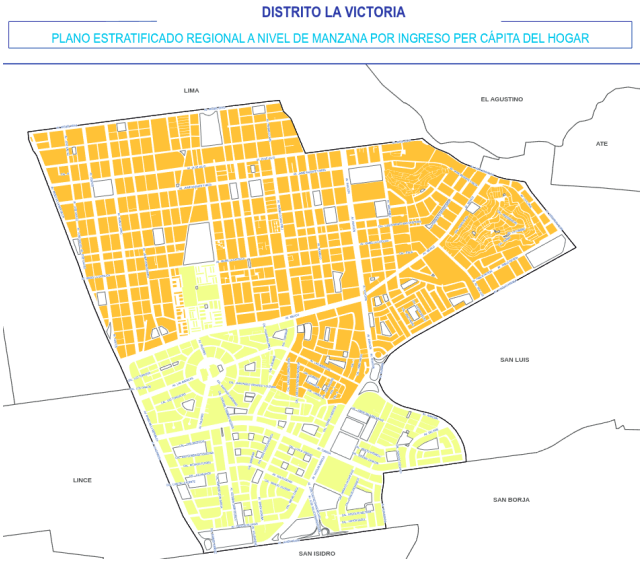

Fig. 1. Socioeconomic stratification of city blocks in La Victoria District.

The INEI study defines five socioeconomic levels: High, Upper Medium, Medium, Lower Medium, and Low. These levels are determined by the per capita income of each household, then averaged across all households within a city block. Each level is represented by a distinct color on the maps, as shown in Table I.

TABLE I
Socioeconomic Level Classification Criteria for the Metropolitan Area of Lima.

SE Level	Color	Per Capita Income (PEN)
High	Purple	$\geq$ 2,412.48
Upper Medium	Yellow	1,449.72 - 2,412.47
Medium	Orange	1,073.01 - 1,449.71
Lower Medium	Brown	803.72 - 1,073.00
Low	Red	$<$ 803.72

For this project, we simplified the five original categories into three broader classes to improve model training and reduce classification complexity:

- High: Corresponds to the original *High* category
- Medium: Combines *Upper Medium* and *Medium* categories
- Low: Combines *Lower Medium* and *Low* categories

We selected 12 districts from the *MAL* based on their socioeconomic homogeneity within each class. This selection ensures that the training data contains representative examples of each category.

A. High category districts

- Miraflores
- San Isidro
- La Molina
- San Borja

These districts have the highest concentration of blocks classified as *High* in the INEI study, with minimal presence of lower socioeconomic levels.

B. Medium category districts

- La Victoria
- Breña
- Lince
- Los Olivos

These districts show a balanced mix of *Upper Medium* and *Medium* blocks, representing the consolidated Medium-class areas of Lima.

C. Low category districts

- Carabayllo
- San Juan de Lurigancho
- Villa El Salvador
- Villa María del Triunfo

These districts contain the highest proportion of *Lower Medium* and *Low* blocks, typically characterized by informal settlements and lower-income residential areas.

V. Feature Engineering

Feature extraction was performed using DINOv2 [4], a self-supervised vision transformer from Meta AI that transforms street-view images into numerical embeddings.

DINOv2 is a pre-trained model that learns visual representations from unlabeled images without requiring manual annotations. The model is based on Vision Transformers (ViT) [5], which divide images into patches and process them through transformer layers to extract meaningful features.

Unlike traditional supervised models, DINOv2 uses self-supervised learning, where the model learns by solving pretext tasks that require understanding image structure. This approach allows it to capture both low-level features (textures, colors) and high-level semantic concepts (objects, scenes, spatial layouts).

For our project, we initially used the pre-trained DINOv2-base model with ViT-B/14 architecture, which outputs 768-dimensional feature vectors. However, to improve classification performance, we experimented with the larger DINOv2-giant model with ViT-g/14 architecture, utilizing CUDA acceleration to handle the higher demands. The giant model takes 224×224 pixel images as input and outputs 1536-dimensional feature vectors, which provided significantly richer representations and improved classification accuracy. To perform the feature extraction with DINOv2, we first resized the images to 224x224, as the images received from the Street View API came in 640x640 size.

The resulting feature vectors serve as input to the four classification models described in the next section.

VI. Selected Models

A. Logistic Regression

Logistic Regression is a linear classification model that estimates class probabilities using the sigmoid function [6]:

$$P(y=1|x) = \frac{1}{1+e^{-(w^T x+b)}}$$

For multiclass problems, the One-vs-Rest (OvR) strategy trains K independent binary classifiers, where each classifier k distinguishes class k from all other classes:

$$f_k(x) = w_k^T x + b_k, \quad k = 1, \ldots, K$$

The final prediction is:

$$\hat{y} = \arg\max_k f_k(x)$$

The regularization parameter C controls model complexity, with smaller values providing stronger regularization.

B. Support Vector Machines (SVM)

SVM finds the optimal hyperplane that maximizes the margin between classes [7]. For non-separable data, the soft-margin formulation introduces slack variables:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to: } y_i(w^T x_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0$$

where C controls the trade-off between margin maximization and training error. For multiclass classification, we use the One-vs-Rest strategy. We employed the RBF (Radial Basis Function) kernel to capture non-linear patterns in the feature space:

$$K(x_i, x_j) = \exp\left(-\gamma\|x_i - x_j\|^2\right)$$

C. Multilayer Perceptron (MLP)

MLP is a feedforward neural network with multiple layers of neurons [8]. For a network with L layers, the forward pass computes:

$$h^{(l)} = f\left(W^{(l)}h^{(l-1)} + b^{(l)}\right), \quad l = 1, \ldots, L$$

where $h^{(l)}$ is the activation of layer l, $W^{(l)}$ and $b^{(l)}$ are weights and biases, and $f(\cdot)$ is the activation function. We use ReLU activation for hidden layers:

$$\text{ReLU}(z) = \max(0, z)$$

The output layer uses softmax activation for multiclass classification. The network is trained by minimizing cross-entropy loss using backpropagation and the Adam optimizer.

D. XGBoost

XGBoost (Extreme Gradient Boosting) is an ensemble learning method that builds an additive model of decision trees through gradient boosting [9]. The model prediction is the sum of predictions from K trees:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in \mathcal{F}$$

where $\mathcal{F}$ is the space of regression trees. At each iteration t, a new tree f_t is added to minimize the objective function:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t)$$

where l is the loss function and $\Omega(f_t)$ is a regularization term:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

Here, T is the number of leaves in tree f_t, w_j are leaf weights, and γ and λ are regularization parameters that control model complexity. Using second-order Taylor expansion of the loss function, the objective at iteration t can be approximated as:

$$\tilde{\mathcal{L}}^{(t)} \approx \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t)$$

where $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ and $h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ are the first and second-order gradients of the loss function. For our three-class problem, XGBoost uses the softmax objective with multiclass logloss. Key hyperparameters include learning rate (η), maximum tree depth, minimum child weight, and column sampling ratio.

VII. Exploratory Data Analysis (EDA)

The Exploratory Data Analysis aims to verify whether the dataset contains consistent visual cues that can support the training of a reliable classification model.

We first examine the distribution of images across classes to detect potential imbalances. Then, we analyze image quality aspects such as resolution, format uniformity, and the presence of corrupted or noisy samples. Additionally, we inspect visual patterns that may differentiate socioeconomic levels, including vegetation presence, street density, architectural uniformity, and illumination conditions. This analysis allows us to validate the viability of the problem, detect outliers, and justify preprocessing decisions such as normalization, augmentation, and class balancing for the subsequent modeling stage.

A. Dataset Structure and Quality Assessment

We first inspected the directory structure of the dataset to confirm that the images were correctly organized into the three socioeconomic classes (High, Medium, Low). All folders were detected without inconsistencies, and non-image directories were properly excluded.

Next, we computed the number of images per class to verify whether the dataset presented any significant imbalance. As shown in Figure 2, the distribution is relatively uniform, with each class containing between 4,000 and 5,500 samples. This balance is beneficial for model training, as it reduces the need for oversampling or class-weight adjustments.

We also performed a validation step to detect corrupted or unreadable files by attempting to load every image in the dataset. No invalid files were found, indicating that the dataset is clean and ready for processing.

Finally, we evaluated the resolution of all images to ensure uniformity across classes. The width and height distributions showed negligible variation, with nearly all images sharing the same dimensions. This consistency simplifies preprocessing and eliminates the need for resizing operations beyond standard normalization for DINOv2 feature extraction.

Fig. 2. Number of images per socioeconomic class.

B. Visual Exploration by Socioeconomic Class

To generate an initial understanding of the visual patterns present in the dataset, we randomly sampled multiple images from each socioeconomic class (High, Medium, Low) and arranged them into mosaics for qualitative inspection. This step allows us to identify recurring characteristics in the urban landscape that may serve as discriminative cues for the classification models.

1) Image Mosaics: Figures 3, 4, and 5 show representative mosaics for each socioeconomic class. Each mosaic contains samples obtained from different districts and highlights the diversity of scenes within each label.

2) Qualitative Observations: The visual inspection reveals consistent structural and aesthetic differences between classes:

Low socioeconomic class Images frequently show unpaved or deteriorated streets, exposed brick walls, informal or unfinished constructions, limited vegetation, and a predominance of dusty or desaturated color tones. The urban layout tends to be irregular, with a high density of cables, informal signage, and heterogeneous building materials.

Medium socioeconomic class: Streets are mostly paved and better maintained, and buildings show more uniformity in height and finishing. Vegetation is present in moderate amounts, road markings are more consistent, and the overall scene exhibits less structural variability compared to the low class.

High socioeconomic class: These areas feature well-maintained roads, planned urban layouts, abundant vegetation, modern facades, and homogeneous architectural styles. The scenes include fewer exposed cables and visual obstructions, resulting in images with cleaner textures and more saturated colors.

3) Summary of Findings: The clear visual differences across socioeconomic classes suggest that street-level images contains discriminative features that can support the supervised

learning task. This qualitative analysis supports the viability of using computer vision models and motivates subsequent quantitative analyses such as color distribution, texture patterns, and feature-space visualization.

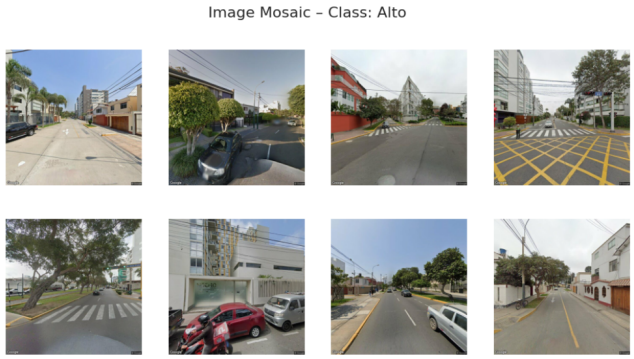

Fig. 3. Representative mosaic of images from the High socioeconomic class.

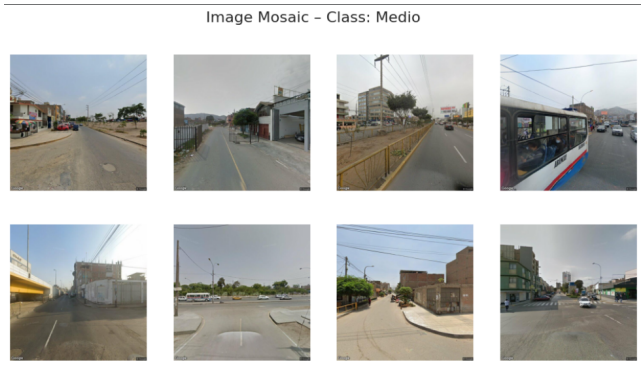

Fig. 4. Representative mosaic of images from the Medium socioeconomic class.

Fig. 5. Representative mosaic of images from the Low socioeconomic class.

C. Color Analysis (RGB)

To evaluate whether chromatic information contributes to the separation of socioeconomic classes, we computed the average RGB values for all images and compared their distributions across classes. Table II summarizes the mean intensity of each channel per class, while Figure 6 shows the joint RGB distribution in a three-dimensional space.

TABLE II
AVERAGE RGB VALUES PER SOCIOECONOMIC CLASS.

Class	R	G	B
High	135.38	135.58	128.42
Medium	136.67	136.18	130.90
Low	147.46	144.02	135.51

The results indicate that the Low class exhibits the highest values across all channels, suggesting higher overall brightness, likely due to unpainted facades, exposed terrain, and increased solar exposure. In contrast, the High and Medium classes show very similar color profiles, with no distinctive chromatic patterns between them.

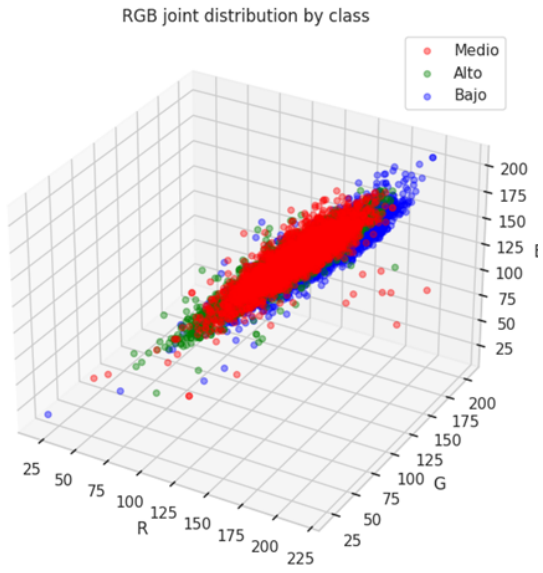

Fig. 6. Joint RGB distribution per socioeconomic class.

Overall, the analysis shows that color alone is insufficient as a discriminative feature for socioeconomic classification. While the Low class is slightly brighter on average, the High and Medium classes overlap extensively in RGB space. Thus, more complex visual cues—such as textures, edges, and high-level semantic features extracted via DINOv2—are required to achieve robust classification performance.

D. Texture Analysis (Edge Density)

To evaluate structural complexity in the images, we computed edge density using the Canny detector. This metric reflects the proportion of pixels identified as contours and serves as a proxy for urban texture (e.g., architectural detail, road markings, vegetation boundaries, and facade geometry). Figure 7 shows the distribution of edge density across the three socioeconomic classes.

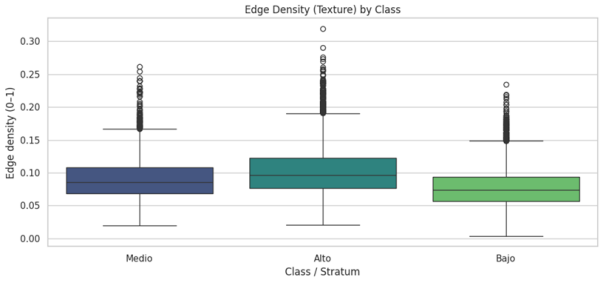

Fig. 7. Edge-density distribution per socioeconomic class.

The results reveal a clear trend: High-class areas show the highest edge density, followed by Medium, while Low-class areas exhibit the lowest values. This suggests that wealthier districts contain more structured and well-defined visual elements (e.g., road markings, modern facades, shadows, vegetation), whereas lower-income districts display flatter surfaces, fewer geometric patterns, and reduced structural detail.
Overall, texture provides a more discriminative signal than color for differentiating socioeconomic classes, supporting the use of convolutional models and high-level feature embeddings in subsequent stages of the project.

E. Dimensionality Reduction on DINOv2 Embeddings

To analyze the latent structure of the dataset, we apply dimensionality reduction techniques on the feature vectors generated using the DINOv2 model. This allows us to evaluate whether the three socioeconomic classes (*High*, *Medium*, *Low*) exhibit any natural separability prior to supervised learning.

1) PCA Analysis: Principal Component Analysis (PCA) was first applied to obtain a linear projection of the embeddings. The objective is twofold: (1) evaluate whether a linear subspace can separate the classes, and (2) visualize the global structure of the data in low-dimensional spaces.

a) PCA 2D Projection: Figure 8 shows the projection of the embeddings onto the first two principal components. The three classes exhibit strong overlap, and no clear linear separation is observed. PCA 2D provides a coarse global overview but is insufficient to capture meaningful class separability.

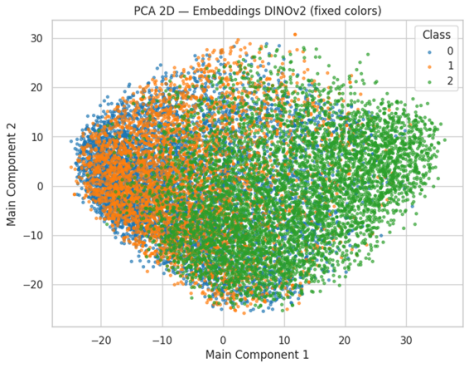

Fig. 8. 2D PCA projection of DINOv2 embeddings.

b) PCA 3D Projection: Adding a third principal component improves visualization richness (Figure 9). Although slight tendencies toward grouping appear, the three classes still form a continuous cloud without distinct clusters. This confirms that the underlying structure of the embeddings is highly complex and not easily represented through linear projections. PCA 3D reveals mild structure but still does not produce clear class boundaries.

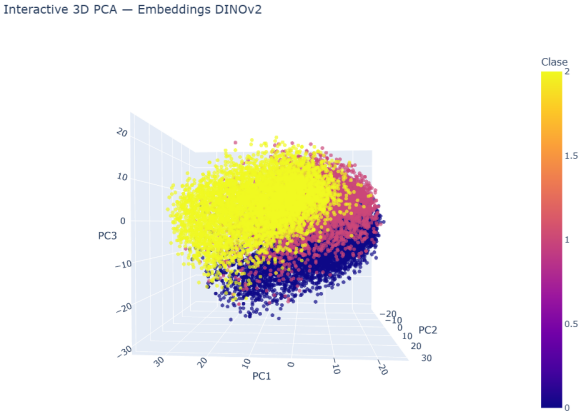

Fig. 9. 3D PCA projection of DINOv2 embeddings.

c) Explained Variance: Scree Plot: Figure 10 presents the cumulative explained variance for the first 50 principal components. The plot shows that:

- the first two components explain less than 15% of variance,
- approximately 50 components are required to surpass 60% variance,
- no clear "elbow" is present in the curve.

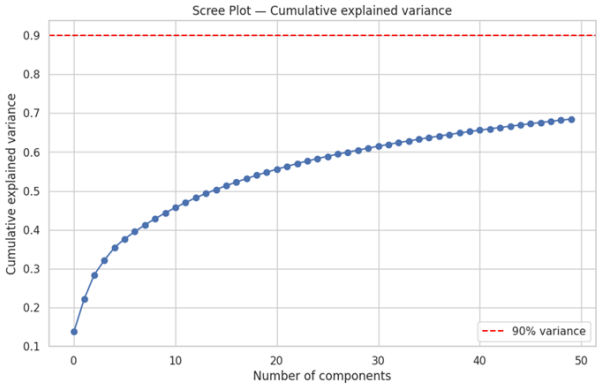

Fig. 10. Cumulative explained variance of PCA components.

d) General Conclusions of PCA: The PCA analysis reveals that:

1) The dataset is not linearly separable.
2) Linear methods such as PCA are not sufficient to identify socioeconomic clusters.
3) More expressive non-linear techniques (t-SNE, UMAP, ISOMAP) are required for structure discovery.

2) t-SNE Nonlinear Reduction: t-SNE is a nonlinear dimensionality reduction technique that preserves local neighborhoods and reveals complex structures in high-dimensional spaces. Unlike PCA, which captures only linear variance, t-SNE focuses on maintaining local relationships, making it effective for exploring clustering tendencies in dense embedding spaces such as those produced by DINOv2.

a) t-SNE 2D: The two-dimensional projection reveals partially defined regions for each socioeconomic class:

- Class 0 tends to concentrate toward the upper-left region.
- Class 1 appears highly dispersed and overlaps with both extremes.
- Class 2 occupies the lower-right area.

Although t-SNE 2D exhibits some structure, the overlap between classes indicates that the separability is limited when restricted to two dimensions.

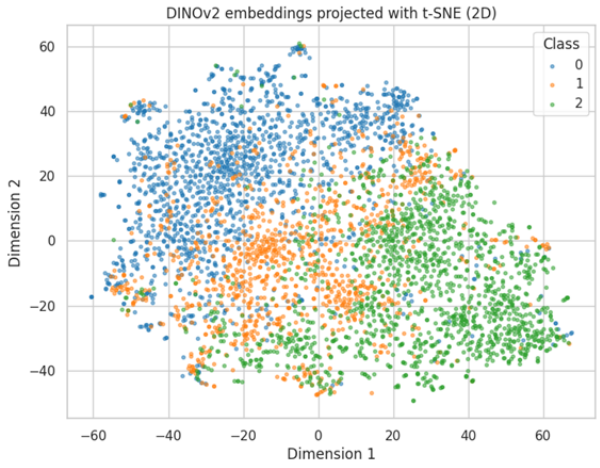

Fig. 11. t-SNE 2D projection of DINOv2 embeddings.

b) t-SNE 3D : Projecting into three dimensions reveals more pronounced patterns:

- Class 0 occupies another distinct region, though with some overlap.
- Class 1 remains the least structured, positioned between the other two classes.
- Class 2 forms a relatively compact group.

The 3D projection provides clearer separation than the 2D version, highlighting nonlinear relationships encoded within the embeddings.

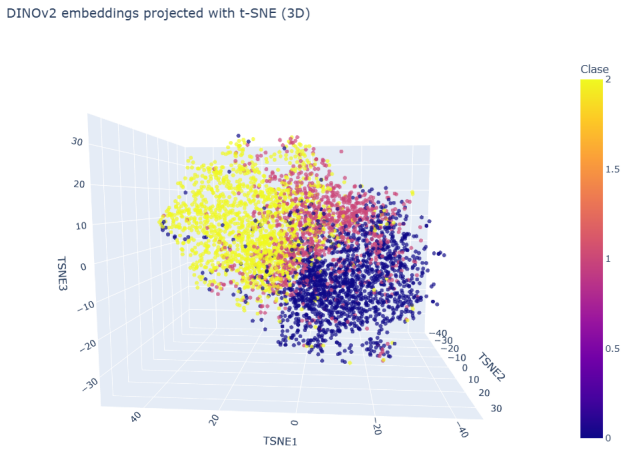

Fig. 12. t-SNE 3D projection of DINOv2 embeddings.

c) Overall Conclusion on t-SNE: t-SNE uncovers partially differentiated clusters within the DINOv2 embedding space, indicating that the socioeconomic classes exhibit nonlinear patterns that are not captured by linear methods like PCA. While the separation is not definitive, the results confirm that the embeddings contain meaningful structure and justify the use of nonlinear dimensionality reduction and advanced classification models.

3) UMAP Nonlinear Reduction: UMAP is a nonlinear dimensionality reduction method that preserves both local and global structure of the high-dimensional manifold. Compared to t-SNE, UMAP tends to produce more stable and coherent separations, making it well-suited for analyzing the organization of DINOv2 embeddings.

a) UMAP 2D (Interpretation): The two-dimensional projection reveals a clearer separation between the socioeconomic classes compared to PCA and t-SNE:

- Class 0 clusters predominantly toward the right region of the plane.
- Class 1 forms a compact cluster on the left.
- Class 2 appears between both extremes, exhibiting intermediate behavior consistent with its socioeconomic position.

Overall, UMAP 2D effectively captures both global and local structure in the embedding space, producing significantly more defined groupings.

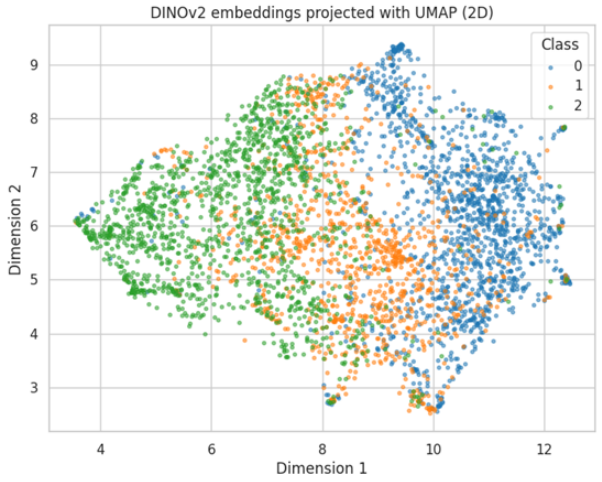

Fig. 13. UMAP 2D projection of DINOv2 embeddings.

b) UMAP 3D (Interpretation): The three-dimensional visualization reinforces the patterns observed in 2D:

- Class 0 occupies another distinct area.
- Class 1 continues to bridge both groups, positioned between their respective clusters.
- Class 2 maintains a well-delimited region.

UMAP 3D provides the strongest class differentiation among all dimensionality reduction techniques tested, revealing a more pronounced latent structure.

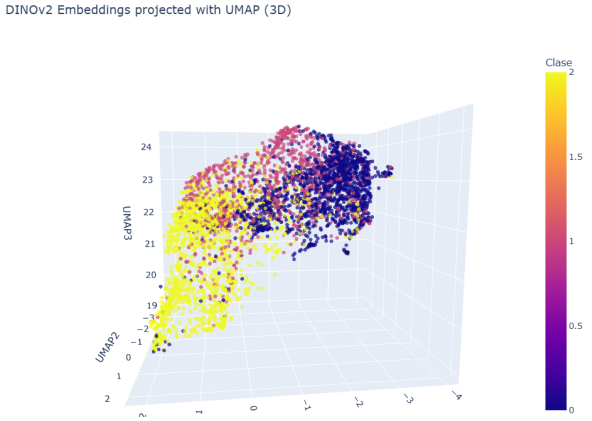

Fig. 14. UMAP 3D projection of DINOv2 embeddings.

c) Overall Conclusion on UMAP: UMAP is the dimensionality reduction method that best separates the classes in the DINOv2 embedding space. Its ability to capture nonlinear relationships in a stable and interpretable manner reveals three partially differentiated groups, indicating that socioeconomic strata contain meaningful visual structure. These findings support the use of deep learning models and high-quality embeddings for the classification task.

4) ISOMAP Nonlinear Reduction: ISOMAP is a nonlinear dimensionality reduction method that models the underlying manifold by approximating geodesic distances between neighboring points. This makes it suitable for datasets distributed along curved or non–linear structures, such as DINOv2 embeddings.

a) ISOMAP 2D (Interpretation): The two-dimensional projection reveals a more organized structure than PCA, and partially comparable to UMAP:

- Class 0 forms a compact cluster on the left side of the projection.
- Class 1 is distributed between both extremes, acting as a transitional region.
- Class 2 appears on the right with greater dispersion.

Overall, ISOMAP 2D indicates that the embedding manifold follows a continuous trajectory where the three socioeconomic classes align along a smooth gradient.

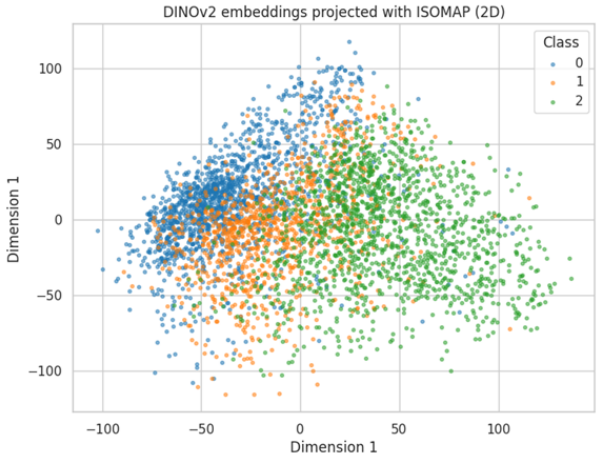

Fig. 15. ISOMAP 2D projection of DINOv2 embeddings.

b) ISOMAP 3D (Interpretation): The three-dimensional projection provides a clearer view of the nonlinear manifold:

- Class 0 concentrates in one extreme of the 3D space.
- Class 1 forms a smooth transition between the two.
- Class 2 occupies the opposite extreme.

ISOMAP 3D captures the curvature of the latent space more clearly than PCA, illustrating how the socioeconomic classes follow a continuous geodesic structure.

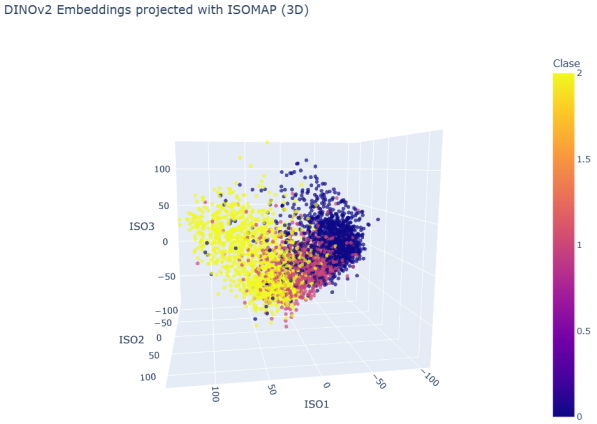

Fig. 16. ISOMAP 3D projection of DINOv2 embeddings.

c) Overall Conclusion on ISOMAP: ISOMAP effectively captures nonlinear relationships in the DINOv2 embedding space. It reveals a continuous, ordered structure in which Class 0 and Class 2 lie at opposite ends, with Class 1 positioned naturally between them. These results suggest that the embeddings encode a meaningful socioeconomic gradient along the manifold.

5) General Conclusions on Dimensionality Reduction: Across all methods, the results indicate that the DINOv2 embeddings contain discriminative socioeconomic information, but it is organized along a continuous nonlinear manifold rather than forming well-separated clusters.

- PCA: Captures global variance but fails to separate classes, indicating that differences are not linearly structured.
- t-SNE: Reveals soft clustering patterns with substantial overlap, suggesting weak but present nonlinear signals.
- UMAP: Provides the clearest visual separation, organizing classes along a continuous gradient in the embedding space.
- ISOMAP: Highlights the curved geometry of the data, showing that classes lie along a smooth socioeconomic trajectory.

Overall, the embeddings encode meaningful socioeconomic variation, but supervised models are required to learn the complex nonlinear boundaries needed for accurate classification.

F. General Conclusion of the EDA

The exploratory analysis shows that, despite limited variation in raw image color and texture, the DINOv2 embeddings capture enough visual information to reflect socioeconomic differences across classes. These differences do not form distinct clusters; instead, they follow a continuous and nonlinear structure in the embedding space.

Dimensionality reduction methods confirm this behavior: PCA reveals no clear separation, while t-SNE, UMAP, and ISOMAP expose gradual transitions and partial groupings. Consequently, the classification task requires a model capable of learning complex nonlinear decision boundaries, leveraging the subtle but meaningful patterns encoded in the embeddings.

VIII. Hyperparameters Analysis

For every model, we used the precalculated features with DINOv2 and split training and test data into 80% and 20% from dataset respectively.

A. Logistic Regression

For this model, we built a pipeline that applies feature normalization using *StandardScaler* and One-vs-Rest (OvR) strategy. This ensures that variables are comparable before training.

To tune hyperparameters, we decided to prove with parameter $C \in \{0.01, 0.1, 1, 10\}$ using L2 penalty. The optimization was done with 5-fold cross validation. In this way we improve model while mitigate overfitting.

The training model was evaluated over 20 different combinations, and the best performing was with $C = 0.01$ and L2. This result suggests that small C improves classification task. This model achieved a cross-validation accuracy of 89.14% and when we evaluated with test data, we got 89.66%. The next confusion matrix shows precision for each class.

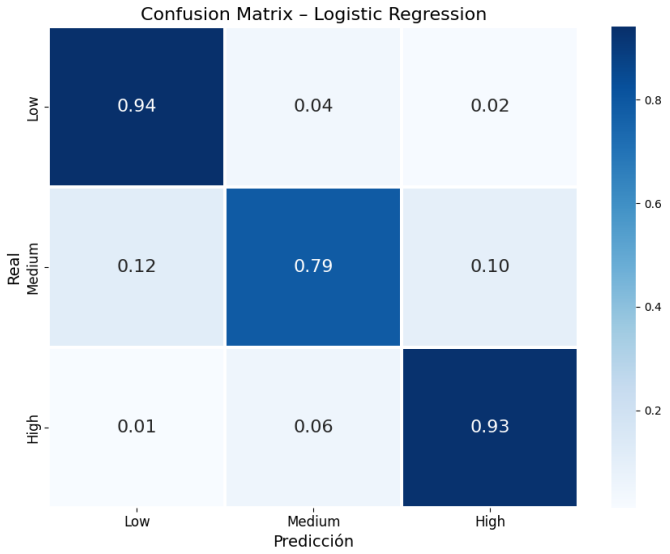

Fig. 17. Logistic Regression, confusion matrix

The classifier shows strong performance on the *High* and *Low* with 0.94 and 0.93 of precision classes, while the *Medium* class remains the most challenging with 0.79 due to class overlap.

B. SVM

To find the best hyperparameters for the SVM model, we first built a pipeline consisting of a *StandardScaler* and an *SVC*. We then defined a distribution of hyperparameters from which we would randomly select some to train the model and calculate the accuracy. This was done using *RandomizedSearchCV*.

The value for *C* was chosen from a uniform distribution in logarithmic scale ranging between 10^{-2} and 10^{2}. The value for *gamma* was chosen from the same distribution, in this case ranging from 10^{-4} to 1. The kernel used was the RBF kernel.

The *RandomizedSearchCV* then selected hyperparameters at random 50 times, performing a 5-fold cross validation for each and calculating the accuracy as the score. At the end it selected the model that resulted in the best score.

The best result included a value of 2.69 for *C* and a value of 0.00048 for *gamma*. This produced a validation accuracy of 0.9085, and after running the model on the test data, we obtained the following confusion matrix:

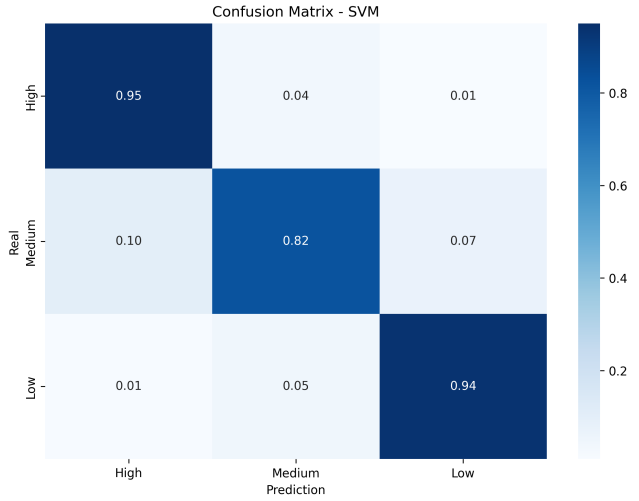

Fig. 18. SVM, confusion matrix

The overall results obtained were: a precision of 0.9106, a recall of 0.9113, an f1-score of 0.9107, and an accuracy of 0.9113.

As such, the model showed great results when predicting high and low classes, with a recall of 0.95 and 0.94 respectively. Regarding the medium classification, we obtained a lower recall of 0.82. This means that the classifier was able to very reliably differentiate between the high and low socioeconomic classifications, but struggled more when trying to decide if it belonged in the medium class. However, the results obtained for this class were considerably accurate, and the overall f1-score and accuracy were also high, meaning that the SVM model performed well.

C. MLP

For this model, we have three hidden layers, the input dimensionality is 1536 because it corresponds the number of extracted features for each image and the output layer contains three units for each class. The model outputs the probability distribution for an image to belong in each class.

During training, we apply K-Fold Cross-Validation over training data using $k = 3$ folds. In this way we reduce overfitting and provides a more reliable estimation for hyperparameters.

We explore every possible combination as:

- dropout: $\{0.2\}$ used to deactive randomly some neurons during training.
- hidden: $\{[1024, 512, 256], [768, 384, 192]\}$ used as the number of neurons on each hidden layer.
- lr: $\{10^{-3}, 5 \times 10^{-4}\}$ controls the step size of optimizer during weight updates.
- batch_size: $\{256, 512\}$ the batch size to process before update weights of model.

In total, 8 models were trained, each corresponding to a unique combination of hyperparameters. The best model that we have chosen is the one who has the minimum loss. Although performance across configurations didn't vary drastically; the best hyperparameters were *dropout*: 0.2, *hidden*: [768, 384, 192], *lr*: 0.001 and *batch_size*: 256.

This model achieves overall accuracy of 89.93%, indicating robust predictive performance.

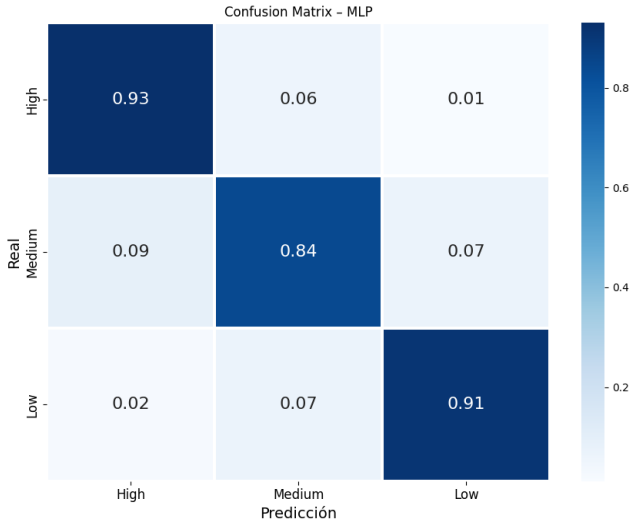

Fig. 19. MLP, confusion matrix

The model achieves consistently strong F1-scores across all classes (around 0.9), showing that its performance is well balanced. Precision is particularly high for the *High* and *Low* classes, obtaining 0.93 and 0.91 respectively. Although the *Medium* class is more difficult to distinguish, the model still obtains a solid precision of 0.84. These results indicate that the classifier is able to reliably differentiate between the three socioeconomic categories.

D. XGBoost

For this experiment, we used the eXtreme Gradient Boosting (XGBoost) classifier, a tree-based ensemble method known for its speed and strong performance on structured data. Features were standardized using *StandardScaler* before training.

To tune the model, we explored a small grid of hyperparameters:

- *max_depth*: $6, 8$, controlling how deep each tree can grow.
- *learning_rate*: 0.1, defining the step size for each boosting update.

The grid is small because training one model took around 45 - 60 minutes, so we couldn't iterate with more hyperparameters. All configurations were trained with 250 estimators (number of trees) and 80% subsampling of rows and features to improve generalization. After evaluating the grid, the best model was obtained with *max_depth*=6 and *learning_rate*=0.1.

This configuration achieved the lowest validation loss, so we retrained it using the full training set and then evaluated it on the test data. The final model reached an accuracy of 88.31%. The resulting confusion matrix is shown below.

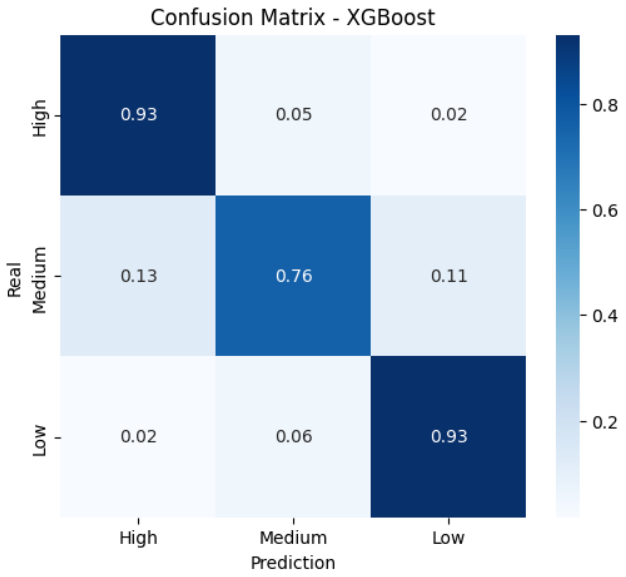

Fig. 20. XGBoost, confusion matrix

Overall, the classifier performs over all categories. It shows high recall for the *High* and *Low* classes (0.89 and 0.90 respectively), while *Medium* class remains the most difficult to classify, with an f1-score of 0.79. This pattern is consistent with the other models and likely reflects overlap between classes in the feature space.

IX. Results

TABLE III
Classification results for the four models.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.8966	0.8900	0.8867	0.8867
SVM	0.9113	0.9106	0.9113	0.9107
MLP	0.8993	0.9000	0.8993	0.8995
XGBoost	0.8800	0.8767	0.8767	0.8800

The above table shows the Accuracy, Precision, Recall and F1-Score metrics for each model. Note that when comparing by F1-Score, the performance of the models, from best to worst is: SVM, MLP, Logistic Regression and XGBoost.

Note that unexpectedly, the base model Logistic Regression is better than XGBoost, which could be explained by the XGBoost simply not performing well. The other models showed better results than the Logistic Regression, which can be explained by the lack of linear separation in the dataset, as shown in the EDA.

The low performance shown by the XGBoost model in comparison to other models can be explained by the fact that there are too many features (1536) in each row, so applying random forest decision did not have positive results and also took too much time during training.

Also, the MLP achieved second place because it would be well for high-dimensional objects and non-linear feature spaces produced by DINOv2. Neural networks can learn complex hierarchical representations, allowing the model to capture subtle interactions between features.

Even though the hyperparameter search was limited, the MLP still managed to generalize well, suggesting that the DINOv2 embeddings contain patterns that benefit from deep architectures rather than shallow or tree-based ones.

The SVM showed the best results. Thanks to the use of an RBF kernel, it was able to separate the classes even when there was no linear separation.

X. Conclusions

A. Main Findings

This study successfully demonstrated that machine learning models combined with pre-trained vision transformers can effectively classify socioeconomic levels in the Metropolitan Area of Lima using Google Street View images. The SVM model with RBF kernel achieved the best performance with 91.13% accuracy, followed by MLP (89.93%) and Logistic Regression (89.66%).

The exploratory data analysis revealed that socioeconomic classes follow a continuous nonlinear manifold structure rather than forming distinct clusters. This finding explains why nonlinear models (SVM with RBF kernel, MLP) outperformed linear approaches.

The Medium class consistently presented the greatest challenge across all models, with F1-scores ranging from 0.79 to 0.85, compared to 0.89-0.95 for High and Low classes. This difficulty stems from the transitional nature of medium-income neighborhoods, which exhibit visual characteristics overlapping with both extremes.

The use of DINOv2 embeddings proved crucial, as its self-supervised learning captured both low-level patterns and high-level semantic concepts. The giant model variant (ViT-g/14) with 1536-dimensional embeddings provided significantly richer representations than the base model.

B. Limitations

- Dataset coverage: The 15,000 images cover only 12 out of 43 districts in Lima, introducing potential selection bias. Informal settlements and peripheral neighborhoods remain underrepresented.
- Temporal validity: The INEI data from 2020 may not perfectly align with more recent street-view images, potentially introducing label noise in rapidly changing neighborhoods.
- Simplified categorization: Merging five original INEI levels into three classes improved performance but sacrificed granularity, particularly for the Medium class which combines distinct socioeconomic subcategories.
- Feature extraction dependency: Using frozen DINOv2 embeddings without fine-tuning may not fully capture Lima-specific visual patterns, as the model was trained primarily on data from developed countries.

C. Future Work

- Expanded geographic coverage: Extending the dataset to all 43 districts would enable better generalization across the entire metropolitan area, with special focus on peripheral and transitional neighborhoods.
- Fine-tuning DINOv2: Adapting the model to Lima-specific visual patterns through careful fine-tuning could improve feature quality and classification performance.
- Multi-modal learning: Combining street-view images with satellite images, OpenStreetMap data, and census statistics could enhance prediction accuracy through complementary information sources.
- Temporal analysis: Collecting images at multiple time points would enable longitudinal studies of socioeconomic change and urban development patterns.
- Transfer learning: Testing models on other Latin American cities would assess generalizability and enable development of region-specific or universal architectures.

REFERENCES

[1] J. Machicao, A. Specht, D. Vellenich, et al., "A deep-learning method for the prediction of socio-economic indicators from street-view imagery using a case study from brazil," *Data Science Journal*, vol. 21, pp. 1–15, 2022.

[2] A. M. Rahman, M. Zaber, Q. Cheng, et al., "Applying state-of-the-art deep-learning methods to classify urban cities of the developing world," *Sensors*, vol. 21, no. 22, p. 7469, 2021.

[3] Instituto Nacional de Estadística e Informática, *Planos Estratificados de Lima Metropolitana a Nivel de Manzanas 2020*. Lima, Perú: Instituto Nacional de Estadística e Informática, 2020. [Online]. Available: https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1744/libro.pdf.

[4] M. Oquab et al., "Dinov2: Learning robust visual features without supervision," *Transactions on Machine Learning Research*, 2024.

[5] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations (ICLR)*, 2021.

[6] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[7] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[8] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[9] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.