Detailed Project Report on Gender Identiﬁcation from Facial Images

**INTRODUCTION**

Gender categorization is an essential visual job for humans since many social interactions rely heavily on gender perception. As visual surveillance and human-computer interaction technologies advance, machines must include an automated and real-time gender identification component. Gender classification systems are used in a variety of applications, including demographic data gathering, pre-processing in face recognition systems, biometrics, and security surveillance systems, as well as other comparable issue areas such as race and ethnicity detection. As a result, gender prediction from face photographs has remained a hot topic in the Computer Vision and Machine Learning communities.

Gender categorization is posed as a multi-class classification issue in which the input face picture is allocated to one of three classes: female, male, or transgender. The steps to address this problem, like other classification problems, may be separated into two phases: feature extraction and classification. Feature extraction entails automatically extracting relevant face characteristics and presenting the input facial picture as a vector of features. Choosing the ideal selection of features to represent the face in an efficient and resilient manner and ensuring that feature extraction occurs in real-time. Our dataset comprises of 50 male, 50 female, and 50 transsexual photos retrieved from Google.

**RELATED WORK**

The first attempts to use computer vision-based approaches for gender categorization were made in the early 1990s. Many other strategies for addressing the challenge of gender identification from face photographs have been documented in the literature since then. The most popular strategy for modeling face processing issues is to represent faces in terms of geometrical aspects. The hidden unit's output was utilized as input to the final neural network, which produced values of 0 for female, 1 for male, and 2 for transsexual.

**METHOD**

Neural Networks

Artificial neural networks (ANNs) are a class of algorithms for statistical learning inspired by neural networks in biology (the central nervous systems of animals, particularly the brain) that are used for estimating or approximate functions that can depend on a large number of inputs and are usually unknown. Layers are commonly used to structure neural networks. Layers are built up of interconnected 'nodes' that each have a 'activation function'. Patterns are supplied to the network via the 'input layer', which communicates to one or more 'hidden layers' where the real working is done via an arrangement of weighted 'connections'. The concealed layers are then connected to a 'output layer'.

Most ANNs include some kind of 'learning rule' that alters the weights of the connections based on the input patterns that it is given. ANNs, like their biological counterparts, learn by example; a youngster learns to recognize dogs by seeing instances of dogs. Although neural networks employ a wide range of learning rules, this presentation focuses just on one: the delta rule. The delta rule is frequently used by the most popular type of ANN, known as 'back propagation neural networks' (BPNNs).

A diagram of a network

Description automatically generated

The architecture of an Artiﬁcial Neural Network

**CONCLUSIONS**

Gender categorization provides the door to a plethora of intriguing applications in related issue fields. It can be used as a pre-processing step in face-recognition software to decrease the search space. More efficient feature extraction approaches can be used to solve this challenge, allowing classification to occur in real or near real time.

CONFUSION MATRIX FOR CLASSIFICATION

[[ 0 1 11] [ 0 2 6] [ 0 0 10]]

**References:**

Hadid, Abdenour, and Matti Pietikäinen. "Demographic classification from face videos using manifold learning." Neurocomputing 100 (2013): 197-205.

Caudill, Maureen. "A Basic Introduction to Neural Networks." Pages. Cs. Wisc. Edu (1989).

**Github Link:**

**https://github.com/DrItauma-NU/exercise-2-deep-learning-with-tensorflow-sishira7.git**