Pseudocode

```
# model: pretrained (e.g. distilled) model
# layer_norm: layer normalization layer
# cls_head: linear classifier -> nn.Linear(D, C)
# x: batch of images (B, 3, H, W)
def image_downstream_forward(model, layer_norm, cls_head, x, linear_probe):

    if linear_probe:
      with torch.no_grad():
        x = model(x) # (B, T, D)
    else:
      x = model(x) # (B, T, D)

    x = x[:, 1:] # remove cls token (B, T-1, D)
    x = x.mean(dim=1) # mean over all patches (B, D)
    x = layer_norm(x)
    x = cls_head(x) # (B, C)
    pred = x.argmax(dim=-1) # (B, )
    return pred
```

Listing 1: Pytorch pseudocode for the forward pass during finetuning or linear probing of a pretrained model on an image classification tasks. The output of the forward pass is the predicted class index for each image in the batch.

```
# teacher_model: ResNet-50-A1 model
# image_encoder: Image encoder of the multimodal student model
# text_encoder: Text encoder of the multimodal student model
# shared_encoder: Shared encoder of the multimodal student model
# imgs: batch of images (B, 3, H, W)
# captions: batch of image captions (B, 64)
# kl_div: KL-Divergence
# clip_loss: Contrastive loss used in CLIP
def forward(teacher_model, image_encoder, text_encoder,
            shared_encoder, imgs, captions):

    with torch.no_grad():
      target = teacher_model(imgs) # (B, 1000)

    img_layer_res = shared_encoder(image_encoder(imgs)[:, 0])
    # [(B, 768), (B, 3072), (B, 1000)]

    text_layer_res = shared_encoder(text_encoder(captions)[:, 0])
    # [(B, 768), (B, 3072), (B, 1000)]

    kl_loss = 1/2*kl_div(target, img_layer_res[2]) +
              1/2*kl_div(target, text_layer_res[2])

    itc_loss = 1/3*clip_loss(img_layer_res[0], text_layer_res[0]) +
               1/3*clip_loss(img_layer_res[1], text_layer_res[1]) +
               1/3*clip_loss(img_layer_res[2], text_layer_res[2])

    loss = kl_loss + itc_loss

    return loss
```

Listing 2: Abstract code used in the forward pass for distilling the multimodal Transformer SHRe from a pretrained ResNet-50-A1 model.